IMAGE TRANSMISSION VIA SPREAD SPECTRUM TECHNIQUES

Robert W. Means, et al

Naval Undersea Center

Prepared for:

Advanced Research Projects Agency

1 October 1973

AD-775234

IMAGE TRANSMISSION
VIA
SPREAD SPECTRUM TECHNIQUES

Investigated by
DR. ROBERT W. MEANS (714:225-6872), Code 608
JEFFREY M. SPEISER (714:225-6607), Code 6021
and
HARPER J. WHITEHOUSE (714:225-6315), Code 6003
Naval Undersea Center
San Diego, California

Sponsored by
Advanced Research Projects Agency
Order Number 2303
Code Number 3G10
Contract
effective: 15 February 1973
expiration: 30 June 1974
amount: $519,400

ARPA Quarterly Technical Report
June 1, 1973 – October 1, 1973
Form Approved Budget Bureau No. 22-R0293

CONTENTS

INTRODUCTION

The feasibility of television bandwidth reduction and spread spectrum transmission has been investigated for a remotely piloted vehicle (RPV) [1]. Two-dimensional transform methods are capable of providing sample rate reductions by zonal encoding or bit rate reductions by combining zonal encoding and block quantization. Differential pulse-code modulation (DPCM) has been established as equivalent to transform encoding when a lower triangular matrix is used to transform the data to a set of uncorrelated signals [2]. Mixed transform coding was investigated and found to be intermediate in performance between the two-dimensional forms of transforms used to encode the horizontal and vertical directions of the image, with the singular exception of the mixed transform composed of an orthogonal transformation and DPCM. For this exception, performance as good as optimum (Karhunen-Loeve transformation and block quantization) was achieved.

Because of the results summarized above, emphasis was placed on investigating transformations which could be easily implemented in real time using compact serial access hardware and would achieve as good a performance as possible while minimizing the amount of memory needed at the transmitter in the RPV. A direct two-dimensional implementation of the Fourier transform was conceived, investigated, and found to be feasible for nearly square arrays as reported in appendix A. Further investigation was made of the mixed transforms using generalized DCPM. Their performance in a noisy channel was evaluated and found to be superior to conventional DPCM systems as reported in appendix B. Surface acoustic wave (SAW) and charge transfer device (CTD) hardware implementations of the Fourier transform were constructed and evaluated and are presented in Appendices C and D. Continued investigation was made of the transform extrapolation techniques and is reported in Appendix E. Further development of the Signal Processing Interpreter is reported in appendix F. A feasibility study of a row-to-column converter is given in appendix G.

SYSTEM ANALYSIS

There are two basically different approaches to the redundancy reduction and spread spectrum transmission of video images for a remotely piloted vehicle: (1) sample reduction through transform encoding and analog transmission; (2) bit rate reduction through transform encoding and block quantization or differential pulse code modulation. The former is the simplest conceptually, and the easiest to implement but imposes restraints on the spread spectrum encoding and on the transmitter. The latter is more difficult to implement but imposes few restrictions on the spread spectrum encoding or on the transmitter and requires buffer memory and quantizers.

In either implementation, transform source encoding is used to redistribute the variance associated with the image samples and then channel encoding is used to provide noise protection and jam resistance. It is at the receiver that the major difference between the two techniques appears. The first is essentially a pulse compression technique, which is possible since the redundancy of the image is removed and spread spectrum redundancy introduced in its place. Thus, the receiver can consist of a simple matched filter matched to the RPV signal, and no synchronization is required between the RPV and the receiving station. The second implementation is essentially a digital communication system in which channel synchronism must be established in order to decode the spread spectrum channel encoding.

The degree of A J protection that can be obtained using the digital encoding system in the RPV is not limited by pulse compression design considerations. The ability to operate with simple hard limiting modems further enhances this alternative. However, some encoding memory will be required in the vehicle and synchronism of the received transmission will be required prior to code demodulation and image reconstruction.

TRANSFORM ENCODING

The capability of transform encoding to provide either a sample rate reduction through zonal encoding or a bit rate reduction through block quantization is well known and documented. [3] The application of these techniques to RPV image bandwidth reduction and spread spectrum encoding is complicated by the large amounts of signal processing required and the need for intermediate memory for two-dimensional transforms. In special cases, these limitations can be circumvented and real time processing is possible with fixed impulse response filters. Previous results established that for a one-dimensional data vector of size N, the Haar and the Fourier transforms could be implemented with, respectively, $\log_2 N + 1$ and 4 real filters.

A new result is obtained. The two-dimensional Fourier transform may be implemented with four real filters and no intermediate memory when the size of the two dimensional transform, N, is the product of the two one-dimensional transform sizes N_1 and N_2, which are relatively prime (i.e., N_1 and N_2 have no common factors). The size of the filters required for implementation are then determined by the transform size N and is either N if recirculating transversal filters are used or 2N-1 if recirculation is not used.

Mixed transforms offer the possibility of improving the performance of two-dimensional transform encoding by combining the implementation advantages of a serial transform in the horizontal direction with the reduction of intermediate memory possible by the use of another transform in the vertical direction. Since the performance is intermediate between the performance achieved by corresponding 2-D transforms of the separate types, the Cosine transform is recommended as the horizontal transform. The vertical transform may then be either the minimum memory implementation of the Haar transform or the generalized DPCM Sample reduction experiments and bit rate reduction comparisons indicate that the combination of the Cosine and the generalized DPCM should be superior in performance to other mixed transforms. For medium error rate channels, the combination of spread spectrum techniques with digital transmission should provide the required output error rates.

TWO-DIMENSIONAL DISCRETE FOURIER TRANSFORM

A class of two-dimensional discrete Fourier transforms (DFT), described in Appendix A are shown to be equivalent to a one-dimensional discrete Fourier transform whose size is the product

of the dimensions of the two-dimensional transforms. This isomorphism allows the two-dimensional DFT to be computed in real time using the serial access implementation of the Chirp Z transform (CZT) with four real filters and parallel implementation of the complex arithmetic. No memory is required to store intermediate results. This may be contrasted with the case when the two-dimensional DFT is implemented by concatination of one-dimensional DFTs and intermediate one-dimensional transforms must be stored prior to transformation in the orthogonal direction. Unfortunately, the input scan required for this two-dimensional DFT is not lexical and thus not directly compatible with conventional raster scanned imagers.

It is shown in Appendix A that the input scan for the two-dimensional DFT is provided by iterated translations on the torus which results by folding the two-dimensional image left to right and top to bottom. This implies that a linearly addressable imager such as a vidicon can be scanned by applying appropriate "saw-tooth" sweeps to both the horizontal and vertical deflection circuits simultaneously. However, for a continuous sweep sensor such as a vidicon, the continuous output signal is not directly the values of the two-dimensional image, and the imager output must be sampled at the appropriate times.

The sampling interval is determined by the dimensions of the two one-dimensional arrays and can be interpreted as the time required to go to the subsequent image point in the two-dimensional space of the input, which may not be adjacent to the preceding point. For the special case when the transform sizes differ by one, then the subsequent sample point is adjacent to the preceding sample point until an edge is encountered, and the scan is a raster on the torus and is inclined at an angle of approximately 45° to the horizontal axis of the original image. The sweep generation to achieve the required sweeps for the two-dimensional array is electronically simple and decomposes into two separate sweep generators, one for horizontal sweep and another for vertical sweep. Each sweep generator can easily be implemented as a recursive loop where the initial condition and sweep increment are determined by the particular block size and aspect ratio chosen.

For a discrete image sensor such as a charge coupled device (CCD) or charge injection device (CID), special attention must be given to the format of output from the image sensor. For the CCD array which has fixed lexical output scan, an additional memory or scan converter is required. For a given two-dimensional block size, this requires approximately the same amount of memory as is needed in the row-to-column scan converter used for the calculation of the DFT by concatination of the two one-dimensional transforms. For the CID image sensor, this

limitation does not apply and an independent horizontal and vertical addressing of the imager is possible. Thus for the CID imager, direct interfacing of the scan generators to the imager is possible and would allow direct implementation of the two-dimensional transform with only three major components: (1) a random access CID array, (2) digital horizontal and vertical scan converters, (3) a serial access CZT transform of size determined by the number of points in the two-dimensional array.

This approach is attractive for small arrays where the number of states required to address the imager is small and the size of the one-dimensional transform is not too large. Current technology would probably limit this approach to arrays of the size 99 X 100 elements or 100 X 101 elements. For larger arrays, the size of the one-dimensional transform would exceed 10^4 and its implementation by transversal filters would be difficult. However, the use of direct implementation of two-dimensional Fourier transforms within blocks can be applied to larger arrays in conjunction with other processing techniques between blocks.

MIXED TRANSFORMS

Mixed transforms can be used either for intraframe encoding or interframe encoding depending on the available memory and the type of mixed transform implemented. Three mixed transforms have been considered for intraframe encoding: (1) the discrete cosine transform (DCT) with the Haar transform, (2) the DCT with generalized DCPM, (3) small block size two-dimensional Fourier with generalized DCPM. In order for a mixed transform to be competitive with one of the conventional two-dimensional transforms, it must offer either superior performance or simplicity of implementation.

Symmetrized transforms which have cosine basis vectors have this property. Of these transforms, the odd length cosine transform appears superior in implementation since it can be implemented as the real part of a CZT and since the transform samples are real and are the samples of an auto correlation function which may then be extrapolated by well known techniques. In simulations of transform performance, the cosine transform has been shown to closely approximate the behavior of the "optimum" Karhunen-Loeve transform. All of these conditions suggest that the cosine transform should be considered as the horizontal transform in a mixed transform implementation.

The minimum memory requirements of the Haar transform and its implementation by both transversal filter and "fast" digital algorithms with a minimum of computational steps makes this

transform attractive as a vertical transform for use with the cosine transform. In some applications, digital implementation of the two-dimensional Haar transform are attractive. In analog systems using mixed transforms where sample rate reduction precedes analog transmission, the zonal filter used for coefficient retention must be asymmetric with less reduction made in the Haar coefficients than in the other transform coefficient.

The benefits of mixed transformation implementation and minimum memory may be achieved for digital transmission by combining a one- or two-dimensional transform with generalized DPCM in a hybrid system. The basis of operation of the hybrid transform is that the transform decorrelates the image within its restraints of transform type, unitary dimensionality, and block size, while the generalized DPCM removes the correlation between transform blocks. This hybrid system is particularly attractive for RPV application since its performance is approximately as good as the "optimum" Karhunen-Loeve transform and its implementation requires minimum memory.

HARDWARE IMPLEMENTATION

When the orthogonal transformation used in an image processing application is implementable by transversal filter techniques, real time operation may be achieved with small, low-power, lightweight hardware. For moderate accuracy transforms (e.g. 7 to 10 bits equivalent input accuracy), sampled data analog transversal filters such as SAW and CTD are attractive.

To investigate the feasibility of these techniques both SAW and CTD Fourier transforms were constructed and evaluated. The SAW transform was a 32 point discrete Fourier transform (DFT) implemented at IF with parallel arithmetic on an ST-cut quartz substrate. The CTD transform was a 100 point modified even cosine transform implemented at video with parallel arithmetic using bucket brigade devices (BBD). Both implementations were successful. Spectrum analysis was performed using both implementations and is reported in appendices C and D.

Comparing the performance capabilities of the two implementations, the development of CCD transform implementation is recommended for the RPV image bandwidth reduction. The CCD will allow operation directly at video frequency with smaller size than a comparable SAW implementation; it also achieves better mass production capabilities and lower production costs.

SAW devices provide an existing technology by which real time video processing via the CZT can be implemented. A 32 point complex DFT implemented with parallel arithmetic and aperiodic convolution is described in Appendix C. Although filters with 512 taps could be constructed on quartz using large area photolithography, a CZT of 50 microseconds of video would require a delay line in excess of 300 mm. This size could be reduced to something in excess of 150 mm by substitution of $Bi_{12}GeO_2$ as the piezoelectric substrate. However, this material is not isopaustic, and delay time compensation is required. Another consideration when using SAW devices for video processing is that these devices are band-pass devices which need to operate at an intermediate frequency (IF) and IF amplifiers and detectors must be used with the SAW device in order to achieve video operation.

If the transmission rate of the processed video is less than the real time video input and minimum memory is a consideration, then transforms of smaller size than 256 or 512 should be considered. A 32 point transform offers a good compromise between transform efficiency which increases with increasing transform size and coefficient storage memory which increases

discontinuously from a fractional line store memory to a frame store memory when the coefficient computation rate exceeds the coefficient transmission rate. Although a frame store memory is possible using recirculating SAW delay lines, this much memory can be organized more efficiently by the addition of specialized control.

With the availability of analog fraime storage memory, it is possible to compute the two-dimensional Cosine or Fourier transform by concatination. If the DFT of the video is computed and stored row by row for a whole frame, then converted to column by column format and the DFT computed on these columns, then the two-dimensional Cosine or Fourier transform results. The feasibility of the required row to column transformations using $Bi_{12}GeO_2$ recirculating delay lines was investigated at Stanford University and is reported in Appendix G.

Under some conditions it is possible to compute the two-dimensional Fourier transform directly using a one-dimensional CZT of large size without the need of row-to-column conversion or other intermediate memory. If the block size of the two-dimensional area, N, is the product of two relatively prime numbers, N_1 and N_2, then an input scan can be found which orders the N two-dimensional data points so that they may be transformed by a one-dimensional transform of size N. This configuration is described in Appendix A. Unfortunately, the two-dimensional cosine transformation can not be computed directly using this reordering, and modification of the video sensor is required in order to provide the required input scan. This technique appears attractive for intermediate block sizes in conjunction with another transform such as DPCM. Under these conditions the transform efficiency of two-dimensional encoding can be combined with moderate one-dimensional CZT implementation.

A BBD with 200 taps was used to construct a CZT approximation to a cosine transform. This component was developed by Texas Instruments for Rome Air Development Center and consists of a cosine and sine BBD on a common silicon substrate. The tap weighting, however, is not correct for an exact cosine transformation, and thus the circuit was used in a spectrum analysis mode as described in Appendix D. These devices operate at a clock rate of several hundred kilohertz and, thus, are suitable for slow scan television processing.

REPORT SUMMARY

This report describes the progress on the second phase of a NUC program on image bandwidth reduction for application to the ARPA RPV problem of sending television images over spread spectrum channels. This report details the first quarter of a hardware development and implementation phase.

Hardware has been constructed to perform linear transformations on the television signal. Both surface acoustic wave and charge transfer devices were built and tested. It was decided to proceed with charge transfer devices for the final design. The study of performance tradeoff as a function of block size and transform has continued. It has been determined that a Cosine transform together with a DPCM transform in the second dimension offers near optimal performance.

A principal result of this quarter is the determination that the DPCM and Cosine mixed transform are an optimal performance system. A second result is that charge transfer device hardware has been built which approximates the final prototype hardware.

REPORT SUMMARY

This report describes the progress on the second phase of a NUC program on image bandwidth reduction for application to the ARPA RPV problem of sending television images over spread spectrum channels. This report details the first quarter of a hardware development and implementation phase.

Hardware has been constructed to perform linear transformations on the television signal. Both surface acoustic wave and charge transfer devices were built and tested. It was decided to proceed with charge transfer devices for the final design. The study of performance tradeoff as a function of block size and transform has continued. It has been determined that a Cosine transform together with a DPCM transform in the second dimension offers near optimal performance.

A principal result of this quarter is the determination that the DPCM and Cosine mixed transform are an optimal performance system. A second result is that charge transfer device hardware has been built which approximates the final prototype hardware.

REFERENCES

1. R. W. Means and H. J. Whitehouse, "Image Transmission vice Spread Spectrum Techniques," ARPA Quarterly Technical Report, June 1973, Order Number 2303, Code Number 3G10.
2. Ibid, Appendix A.
3. A. Habibi and P. A. Wintz, "Image Coding by Linear Transformation and Block Quantization," IEEE Trans. Commun. Technol., Vol. COM-19, Feb. 1971, pp 50-62.

APPENDIX A

TWO-DIMENSIONAL DISCRETE FOURIER TRANSFORM ARCHITECTURE

A TWO-DIMENSIONAL DISCRETE FOURIER TRANSFORM ARCHITECTURE

by

J. M. SPEISER

H. J. WHITEHOUSE

Sensor and Information Technology Department

Naval Undersea Center
San Diego, California 92132

ABSTRACT

Many problems of image processing and other processing of two-dimensional data require the rapid generation of two-dimensional Fourier transforms in a form suitable to subsequent electronic processing or utilization. This paper describes a two-dimensional transform architecture which is suitable for real-time image processing and which may be implemented with small, light-weight special purpose hardware.

INTRODUCTION

Two-dimensional Fourier transforms are ordinarily generated either optically or through the use of a digital computer. Both of these methods suffer from severe limitations in many signal processing applications. A general purpose digital computer large enough to perform two-dimensional Fourier transforms in real time for many signal processing applications is large, heavy, expensive, and consumes a large amount of electrical power. Coherent optical two-dimensional Fourier transform implementations are limited by their input-output media and interface poorly with electronic systems.

A third method for performing two-dimensional Fourier transforms is to perform a row by row one-dimensional transform with serial access hardware such as that described previously [1-3], store the one-dimensional row transforms, and perform a column by column one-dimensional transform. This method, however, requires an auxiliary memory to store the one-dimensional transforms of the entire data field. The auxiliary memory tends to be expensive in terms of cost, weight, and power consumption. It is therefore desirable to seek a method of using the one-dimensional serial-access Fourier transform device with a minimum of auxiliary hardware to perform two-dimensional Fourier transforms.

A TWO-DIMENSIONAL DFT ARCHITECTURE USING ONE-DIMENSIONAL HARDWARE

The two-dimensional discrete Fourier transform (DFT) system will use an input scanning device and a one-dimensional discrete Fourier transform device as shown in Figure 1. The two-dimensional transform block size N_1 by N_2 is chosen such that N_1 and N_2 are relatively prime integers (i.e., they must have no common divisor). The one-dimensional Fourier transform device has a block length of $N = N_1 N_2$.

The purpose of the input scanning device is to so order the input data that the one-dimensional Fourier transform of the length $N_1 N_2$ serial data string is identical to an N_1 by N_2 two-dimensional Fourier transform of the N_1 by N_2 input data samples. If desired, an output scanning device may also be used to provide the transform output points in normal order.

The required scan may be derived from Preisendorfer's [4] representation of a one-dimensional discrete Fourier transform matrix as a direct product matrix. Preisendorfer shows how the one-dimensional DFT in equation (1) is equivalent to the two-dimensional DFT in equation (2) when $N = N_1 N_2$ and N_1, N_2 are relatively prime.

$$\hat{f}_q = \sum_{p=0}^{N-1} f_p \, e^{-i2\pi\left(\frac{pq}{N}\right)} \qquad \text{for } p,q = 0, 1, \ldots N-1 \tag{1}$$

$$\hat{g}(k_1,k_2) = \sum_{j_1=0}^{N_1-1} \sum_{j_2=0}^{N_2-1} g(j_1,j_2) \, e^{-i2\pi\left[\frac{j_1 k_1}{N_1} + \frac{j_2 k_2}{N_2}\right]} \tag{2}$$

$$\text{for } j_1,k_1 = 0, 1, \ldots N_1-1$$

$$j_2,k_2 = 0, 1, \ldots N_2-1$$

In order to make the two transforms equivalent, it is necessary to find a pair of one-to-one functions $P(j_1, j_2)$ and $Q(k_1, k_2)$ such that

$$\frac{PQ}{N} = \frac{j_1 k_1}{N_1} + \frac{j_2 k_2}{N_2} \qquad \text{(Modulo 1)} \tag{3}$$

or

$$PQ = j_1 k_1 N_2 + j_2 k_2 N_1 \qquad \text{(Modulo } N_1 N_2\text{)} \tag{4}$$

This may be accomplished by letting

$$P(j_1, j_2) = j_1 N_2 + j_2 N_1 \qquad \text{(Modulo N)} \tag{5}$$

$$Q(k_1, k_2) = k_1 U_1 N_2 + k_2 U_2 N_1 \qquad \text{(Modulo N)} \tag{6}$$

where the constants U_1 and U_2 are the solutions of

$$N_2 U_1 = 1 \qquad \text{(Modulo } N_1\text{)} \tag{7}$$

$$N_1 U_2 = 1 \qquad \text{(Modulo } N_2\text{)} \tag{8}$$

Equations (7) and (8) will have solutions if and only if N_1 and N_2 are mutually prime. Under this condition, the mappings described by equations (5) and (6) will satisfy the requirement of equation (3).

Equation (5) may be interpreted as a prescription for building the input scan generator. As j_1 ranges from 0 to N_1-1 and j_2 ranges from 0 to N_2-1, P will take all values from 0 to $N_1 N_2-1$. These may be tabulated and rearranged in increasing order.

i.e. $0 = P(j_1(0), j_2(0))$

$1 = P(j_1(1), j_2(1))$

$\vdots$

$N-1 = P(j_1(N-1), j_2(N-1))$

The values $j_1(0), \ldots, j_1(N-1)$ and $j_2(0), \ldots, j_2(N-1)$ may be stored in a pair of read-only memories to provide the input scan generator.

To eliminate the ROM's, we note that

$$p\left(j_1(l) + j_1(s), j_2(l) + j_2(s)\right) = p\left(j_1(s), j_2(s)\right) + p\left(j_1(l), j_2(l)\right)$$

Therefore

$$j_1(s+l) = j_1(s) + j_1(l) \qquad (\text{Mod } N_1)$$

$$j_2(s+l) = j_2(s) + j_2(l) \qquad (\text{Mod } N_2)$$

So that only simple recursive digital filters are needed for the input scan generator of Figure 2. With only a change in the prestored constants, replacing $j_1(l)$ by $k_1(l)$ and $j_2(l)$ by $k_2(l)$, the circuit of Figure 2, will also function as the output scan generator. Representative input and output scans are shown in Figures 3 and 4.

A typical application of the two-dimensional Fourier transform device would be in a reduced redundancy television system, as shown in Figure 5. The selection table and selection gate may be used so that only selected spatial Fourier transform coefficients will be transmitted through the channel.

In addition to eliminating the intermediate transform memory, it is also possible to eliminate the ROM's used in the CZT implementations of the one-dimensional DFT,

Let $W = e^{\frac{-i2\pi}{N}}$

Then the discrete chirps needed by the CZT, $p_K = W^{k^2/2}$, satisfy the recursion relation

$$p_{k+2} = W\, p^2_{k+1}/p_k$$

or

$$p_{k+2} = W\, p^2_{k+1}\, p^*_k$$

This may be implemented by the recursive digital filter shown in Figure 6, in which only the single constant W need be stored.

CONCLUSIONS

As contrasted with previous digital two-dimensional Fourier transform devices, the device described here can be extremely fast since a large number of multipliers are effectively utilized in the correlator or convolver of the one-dimensional DFT it uses. As contrasted with previous optical two-dimensional Fourier transform devices, the device described here interfaces extremely well with electronic signal processing systems, and lends itself to potential mass production via integrated circuit fabrication techniques, representing a potential very large savings in cost for quantity production.

The one-dimensional DFT which is utilized may be implemented using accoustic surface waves, charge coupled devices, digital LSI correlators, or other serial access transversal filters or correlators.

Transforms of higher dimensionality may be implemented by adding additional sections to the scan generator. The only fundamental constraint is that the size, N, of the one-dimensional DFT device of Figure 3 must be the product of the individual dimension sizes $N_1, \ldots, N_p$, and every pair N_R, N_S must be mutually prime.

REFERENCES

1. Whitehouse, H. J., J. M. Speiser, and R. W. Means, High Speed Serial Access Linear Transform Implementations, presented at the All Applications Digital Computer (AADC) Symposium, Orlando, Florida, 23-25 January 1973, reprinted as NUC TN 1026.

2. Means, R. W., D. D. Buss, and H. J. Whitehouse, Real Time Discrete Fourier Transforms Using Charge Transfer Devices, Proceedings of the CCD Applications Conference held at the Naval Electronics Laboratory Center, San Diego, Calif., 18-20 Sept. 1973, pp. 95-101.

3. Alsup, J. M., R. W. Means, and H. J. Whitehouse, Real Time Discrete Fourier Transforms Using Surface Acoustic Wave Devices, to appear in the Proceedings of the IEE International Specialist Seminar on Component Performance and Systems Applications of Surface Acoustic Wave Devices, held at Aviemore, Scotland, 24-28 Sept. 1973.

4. Preisendorfer, R. W., Introduction to Fast Fourier Transforms, Visibility Laboratory, University of California, San Diego, Spring 1967.

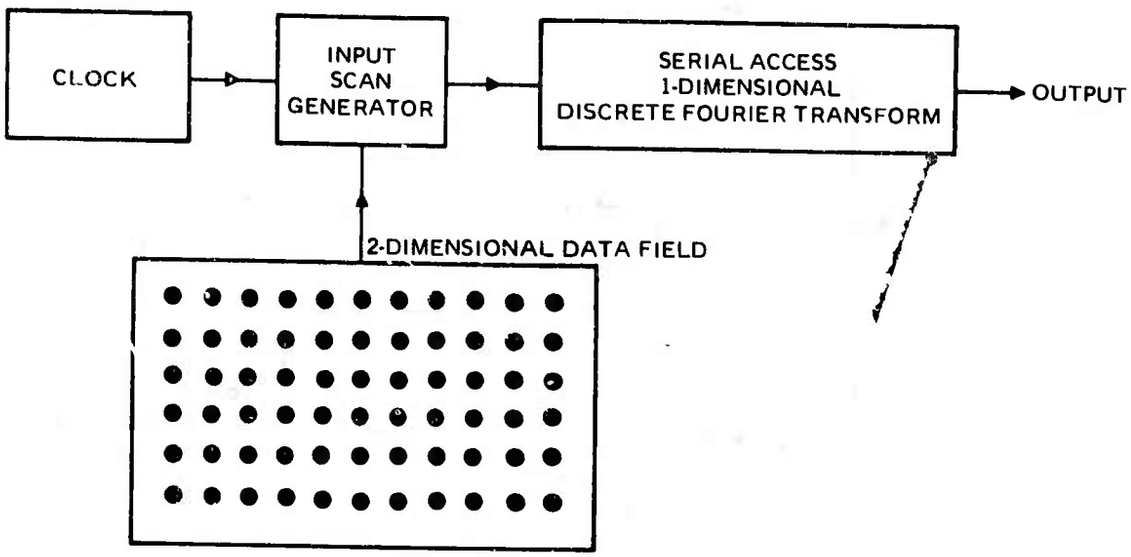


Figure 1. Two-dimensional discrete Fourier transform device.

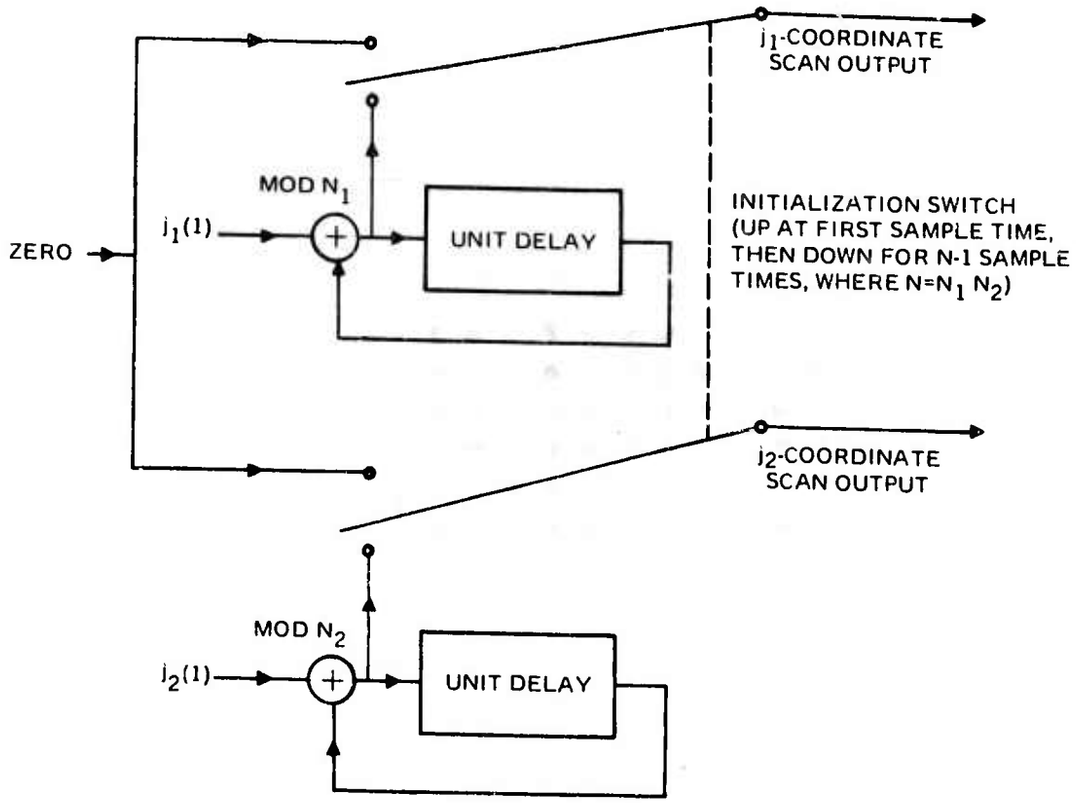


Figure 2. Input scan generator for two-dimensional discrete Fourier transform.

3.1 $N_1 = 3$ $N_2 = 4$ $N = 12$

0	4	8
3	7	11
6	10	2
9	1	5

3.2 $N_1 = 3$ $N_2 = 5$ $N = 15$

0	5	10
3	8	13
6	11	1
9	14	4
12	2	7

3.3 $N_1 = 3$ $N_2 = 7$ $N = 21$

0	7	14
3	10	17
6	13	20
9	16	2
12	19	5
15	1	8
18	4	11

3.4 $N_1 = 3$ $N_2 = 8$ $N = 24$

0	8	16
3	11	19
6	14	22
9	17	1
12	20	4
15	23	7
18	2	10
21	5	13

3.5 $N_1 = 3$ $N_2 = 10$ $N = 30$

0	10	20
3	13	23
6	16	26
9	19	29
12	22	2
15	25	5
18	28	8
21	1	11
24	4	14
27	7	17

Figure 3. Representative Input Scans

3.6 $N_1 = 4$ $N_2 = 5$ $N = 20$

0	5	10	15
4	9	14	19
8	13	18	3
12	17	2	7
16	1	6	11

3.7 $N_1 = 4$ $N_2 = 7$ $N = 28$

0	7	14	21
4	11	18	25
8	15	22	1
12	19	26	5
16	23	2	9
20	27	6	13
24	3	10	17

3.8 $N_1 = 4$ $N_2 = 9$ $N = 36$

0	9	18	27
4	13	22	31
8	17	26	35
12	21	30	3
16	25	34	7
20	29	2	11
24	33	6	15
28	1	10	19
32	5	14	23

3.9 $N_1 = 5$ $N_2 = 6$ $N = 30$

0	6	12	18	24
5	11	17	23	29
10	16	22	28	4
15	21	27	3	9
20	26	2	8	14
25	1	7	13	19

3.10 $N_1 = 5$ $N_2 = 7$ $N = 35$

0	7	14	21	28
5	12	19	26	33
10	17	24	31	3
15	22	29	1	8
20	27	34	6	13
25	32	4	11	18
30	2	9	16	23

Figure 3. Representative Input Scans (Continued)

3.11 $N_1 = 5$ $N_2 = 8$ $N = 40$

0	8	16	24	32
5	13	21	29	37
10	18	26	34	2
15	23	31	39	7
20	28	36	4	12
25	33	1	9	17
30	38	6	14	22
35	3	11	19	27

3.12 $N_1 = 5$ $N_2 = 9$ $N = 45$

0	9	18	27	36
5	14	23	32	41
10	19	28	37	1
15	24	33	42	6
20	29	38	2	11
25	34	43	7	16
30	39	3	12	21
35	44	8	17	26
40	4	13	22	31

3.13 $N_1 = 6$ $N_2 = 7$ $N = 42$

0	7	14	21	28	35
6	13	20	27	34	41
12	19	26	33	40	5
18	25	32	39	4	11
24	31	38	3	10	17
30	37	2	9	16	23
36	1	8	15	22	29

3.14 $N_1 = 7$ $N_2 = 8$ $N = 56$

0	8	16	24	32	40	48
7	15	23	31	39	47	55
14	22	30	38	46	54	6
21	29	37	45	53	5	13
28	36	44	52	4	12	20
35	43	51	3	11	19	27
42	50	2	10	18	26	34
49	1	9	17	25	33	41

Figure 3. Representative Input Scans (Continued)

3.15 $N_1 = 7$ $N_2 = 9$ $N = 63$

0	9	18	27	36	45	54
7	16	25	34	43	52	61
14	23	32	41	50	59	5
21	30	39	48	57	3	12
28	37	46	55	1	10	19
35	44	53	62	8	17	26
42	51	60	6	15	24	33
49	58	4	13	22	31	40
56	2	11	20	29	38	47

3.16 $N_1 = 7$ $N_2 = 10$ $N = 70$

0	10	20	30	40	50	60
7	17	27	37	47	57	67
14	24	34	44	54	64	4
21	31	41	51	61	1	11
28	38	48	58	68	8	18
35	45	55	65	5	15	25
42	52	62	2	12	22	32
49	59	69	9	19	29	39
56	66	6	16	26	36	46
63	3	13	23	33	43	53

3.17 $N_1 = 8$ $N_2 = 9$ $N = 72$

0	9	18	27	36	45	54	63
8	17	26	35	44	53	62	71
16	25	34	43	52	61	70	7
24	33	42	51	60	69	6	15
32	41	50	59	68	5	14	23
40	49	58	67	4	13	22	31
48	57	66	3	12	21	30	39
56	65	2	11	20	29	38	47
64	1	10	19	28	37	46	55

3.18 $N_1 = 9$ $N_2 = 10$ $N = 90$

0	10	20	30	40	50	60	70	80
9	19	29	39	49	59	69	79	89
18	28	38	48	58	68	78	88	8
27	37	47	57	67	77	87	7	17
36	46	56	66	76	86	6	16	26
45	55	65	75	85	5	15	25	35
54	64	74	84	4	14	24	34	44
63	73	83	3	13	23	33	43	53
72	82	2	12	22	32	42	52	62
81	1	11	21	31	41	51	61	71

Figure 3. Representative Input Scans (Continued)

4.1 N = 12 N_1 = 3 N_2 = 4 U_1 = 1 U_2 = 3

0	4	8
9	1	5
6	10	2
3	7	11

4.2 N = 15 N_1 = 3 N_2 = 5 U_1 = 2 U_2 = 2

0	10	5
6	1	11
12	7	2
3	13	8
9	4	14

4.3 N = 21 N_1 = 3 N_2 = 7 U_1 = 1 U_2 = 5

0	7	14
15	1	8
9	16	2
3	10	17
18	4	11
12	19	5
6	13	20

4.4 N = 24 N_1 = 3 N_2 = 8 U_1 = 2 U_2 = 3

0	16	8
9	1	17
18	10	2
3	19	11
12	4	20
21	13	5
6	22	14
15	7	23

4.5 N = 30 N_1 = 3 N_2 = 10 U_1 = 1 U_2 = 7

0	10	20
21	1	11
12	22	2
3	13	23
24	4	14
15	25	5
6	16	26
27	7	17
18	28	8
9	19	29

Figure 4. Representative Output Scans

4.6 N = 20 N_1 = 4 N_2 = 5 U_1 = 1 U_2 = 4

0	5	10	15
16	1	6	11
12	17	2	7
8	13	18	3
4	9	14	19

4.7 N = 28 N_1 = 4 N_2 = 7 U_1 = 3 U_2 = 2

0	21	14	7
8	1	22	15
16	9	2	23
24	17	10	3
4	25	18	11
12	5	26	19
20	13	6	27

4.8 N = 36 N_1 = 4 N_2 = 9 U_1 = 1 U_2 = 7

0	9	18	27
28	1	10	19
20	29	2	11
12	21	30	3
4	13	22	31
32	5	14	23
24	33	6	15
16	25	34	7
8	17	26	35

4.9 N = 30 N_1 = 5 N_2 = 6 U_1 = 1 U_2 = 5

0	6	12	18	24
25	1	7	13	19
20	26	2	8	14
15	21	27	3	9
10	16	22	28	4
5	11	17	23	29

4.10 N = 35 N_1 = 5 N_2 = 7 U_1 = 3 U_2 = 3

0	21	7	28	14
15	1	22	8	29
30	16	2	23	9
10	31	17	3	24
25	11	32	18	4
5	26	12	33	19
20	6	27	13	34

Figure 4. Representative Output Scans (Continued)

4.11 N = 40 $N_1 = 5$ $N_2 = 8$ $U_1 = 2$ $U_2 = 5$

0	16	32	8	24
25	1	17	33	9
10	26	2	18	34
35	11	27	3	19
20	36	12	28	4
5	21	37	13	29
30	6	22	38	14
15	31	7	23	39

4.12 N = 45 $N_1 = 5$ $N_2 = 9$ $U_1 = 4$ $U_2 = 2$

0	36	27	18	9
10	1	37	28	19
20	11	2	38	29
30	21	12	3	39
40	31	22	13	4
5	41	32	23	14
15	6	42	33	24
25	16	7	43	34
35	26	17	8	44

4.13 N = 42 $N_1 = 6$ $N_2 = 7$ $U_1 = 1$ $U_2 = 6$

0	7	14	21	28	35
36	1	8	15	22	29
30	37	2	9	16	23
24	31	38	3	10	17
18	25	32	39	4	11
12	19	26	33	40	5
6	13	20	27	34	41

4.14 N = 56 $N_1 = 7$ $N_2 = 8$ $U_1 = 1$ $U_2 = 7$

0	8	16	24	32	40	48
49	1	9	17	25	33	41
42	50	2	10	18	26	34
35	43	51	3	11	19	27
28	36	44	52	4	12	20
21	29	37	45	53	5	13
14	22	30	38	46	54	6
7	15	23	31	39	47	55

Figure 4. Representative Output Scans (Continued)

4.15 $N = 63$ $N_1 = 7$ $N_2 = 9$ $U_1 = 4$ $U_2 = 4$

0	36	9	45	18	54	27
28	1	37	10	46	19	55
56	29	2	38	11	47	20
21	57	30	3	39	12	48
49	22	58	31	4	40	13
14	50	23	59	32	5	41
42	15	51	24	60	33	6
7	43	16	52	25	61	34
35	8	44	17	53	26	62

4.16 $N = 70$ $N_1 = 7$ $N_2 = 10$ $U_1 = 5$ $U_2 = 3$

0	50	30	10	60	40	20
21	1	51	31	11	61	41
42	22	2	52	32	12	62
63	43	23	3	53	33	13
14	64	44	24	4	54	34
35	15	65	45	25	5	55
56	36	16	66	46	26	6
7	57	37	17	67	47	27
28	8	58	38	18	68	48
49	29	9	59	39	19	69

4.17 $N = 72$ $N_1 = 8$ $N_2 = 9$ $U_1 = 1$ $U_2 = 8$

0	9	18	27	36	45	54	63
64	1	10	19	28	37	46	55
56	65	2	11	20	29	38	47
48	57	66	3	12	21	30	39
40	49	58	67	4	13	22	31
32	41	50	59	68	5	14	23
24	33	42	51	60	69	6	15
16	25	34	43	52	61	70	7
8	17	26	35	44	53	62	71

4.18 $N = 90$ $N_1 = 9$ $N_2 = 10$ $U_1 = 1$ $U_2 = 9$

0	10	20	30	40	50	60	70	80
81	1	11	21	31	41	51	61	71
72	82	2	12	22	32	42	52	62
63	73	83	3	13	23	33	43	53
54	64	74	84	4	14	24	34	44
45	55	65	75	85	5	15	25	35
36	46	56	66	76	86	6	16	26
27	37	47	57	67	77	87	7	17
18	28	38	48	58	68	78	88	8
9	19	29	39	49	59	69	79	89

Figure 4. Representative Output Scans (Continued)

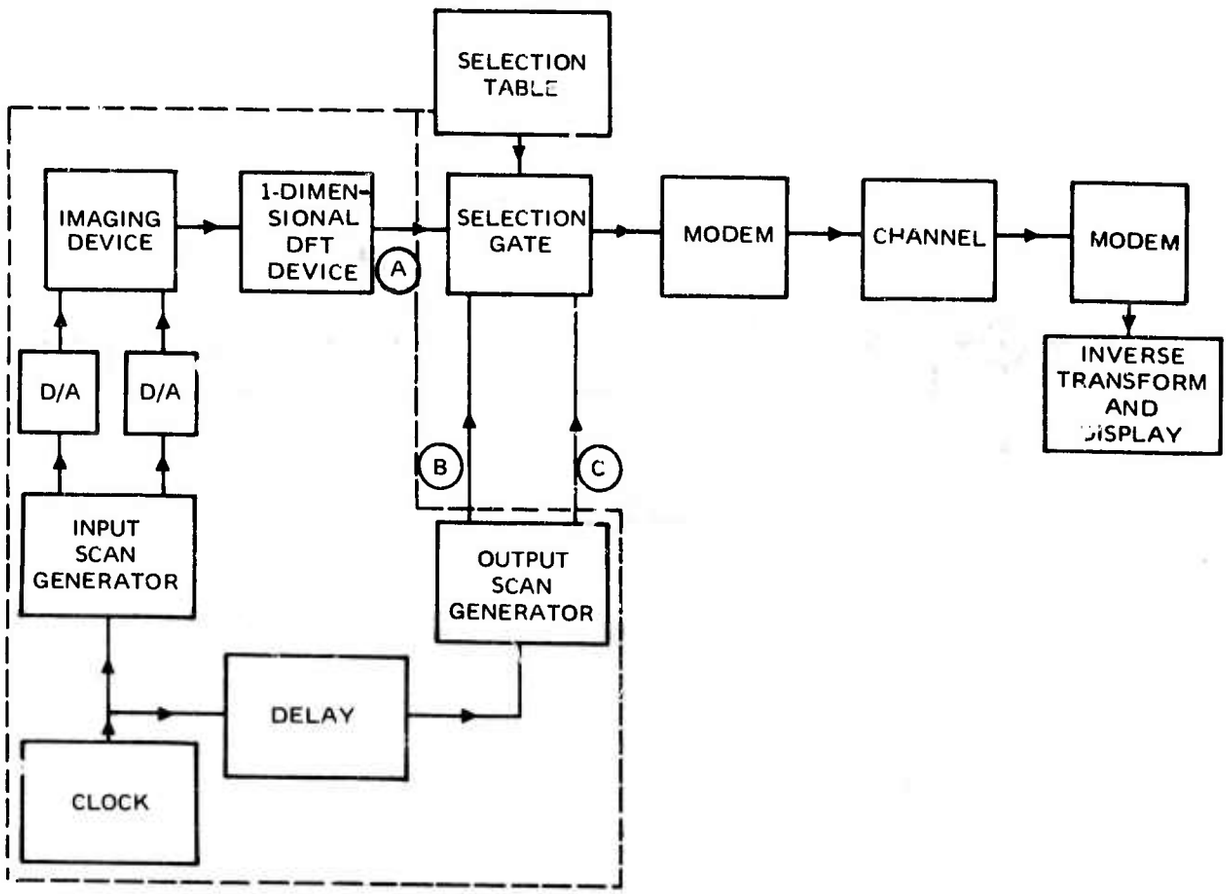


Figure 5. Reduced redundancy image transmission system using two-dimensional discrete Fourier transform device.

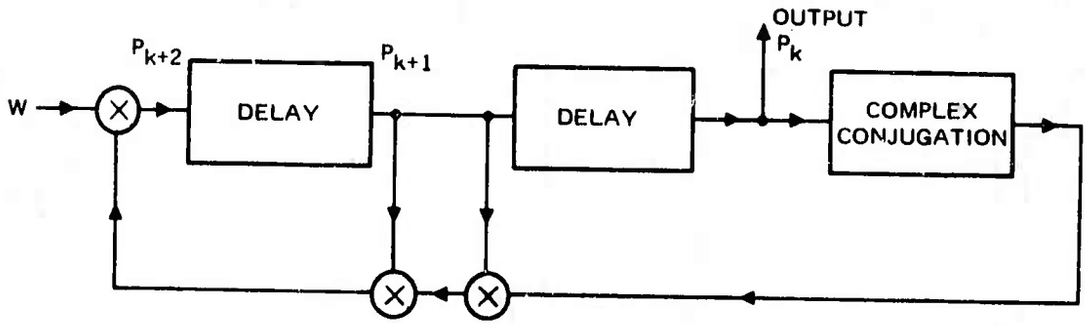


Figure 6. Recursive filter for discrete chirp generation.

APPENDIX B

HYBRID CODING OF PICTORIAL DATA

HYBRID CODING OF PICTORIAL DATA*

by

A. Habibi

Department of Electrical Engineering

University of Southern California

Los Angeles, California 90007

Abstract

Two hybrid coding systems utilizing a cascade of a unitary transformation and DPCM systems are proposed. Both systems encode the transformed data by a bank of DPCM systems. The first system uses a one-dimensional transform of the data where the second one employs two-dimensional transformations. Theoretical results for Markov data and experimental results for a typical picture is presented for Hadamard, Fourier, Cosine, Slant, and the Karhunen-Loeve transformations. The visual effects of channel error and also the impact of noisy channel on the performance of the hybrid system, measured in terms of the signal to noise ratio of the encoder, is examined and the performance of this system is compared to the performances of the two-dimensional DPCM and the standard two-dimensional transform encoders.

This research was supported by the Naval Undersea Center, San Diego, under contract N00123-73-C-1507 and by the Advanced Research Projects Agency of the Department of Defense and was monitored by the Air Force Eastern Test Range under contract F08606-72-C-0008.

I. Introduction

In search for efficient techniques of transmitting pictorial data over digital communication channels has led various researchers to a common approach to the problem. Briefly, this approach is processing the correlated data (images) to generate a set of uncorrelated or as nearly uncorrelated as possible set of signals which in turn are quantized using a memoryless quantizer. The quantized signal is then encoded using either fixed or variable length code words and is transmitted over a digital channel. This is the general approach taken in designing differential pulse code modulators (DPCM) and the techniques that use unitary transformation and block quantization as well as many other techniques developed in recent literature [1-6]. Both DPCM and transform coding techniques have been used with some success in coding pictorial data. A study of both these systems has indicated that each technique has some attractive characteristics and some limitations. The transform coding systems achieve superior coding performance at lower bit rates, they distribute the coding degradation in a manner less objectionable to a human viewer and show less sensitivity to data statistics (picture-to-picture variation) and are less vulnerable to channel noise. On the other hand, DPCM systems when designed to take advantage of spatial correlations of the data achieve a better coding performance at a higher bit rate, the equipment complexity and the delay due to coding operation is minimal, and the system does not require the large memory needed in the transform coding systems. Perhaps the most desirable characteristic of this system is the ease of design and the speed of the operation that has made the use of DPCM systems in real time coding of television signals possible. The limitations of this system are the sensitivity of the well-designed two-dimensional DPCM systems to picture statis-

tics and the propagation of the channel error on the transmitted picture.

In this paper, two coding systems that use a cascade of unitary transformations and a bank of DPCM systems are proposed. These systems combine the attractive features of both transform coding and the DPCM systems thus achieving good coding capabilities without many of the limitations of each system. The first system proposed here exploits the correlation of the data in the horizontal direction by taking a one-dimensional transform of each line of the picture, then it operates on each column of the transformed data using a bank of DPCM systems. The DPCM systems quantize the signal in the transform domain where it takes advantage of the vertical correlation of the transformed data to reduce the coding error. The unitary transformation involved is a one-dimensional transformation of individual lines of the pictorial data, thus the equipment complexity and the number of computational operations is considerably less than what is involved in a two-dimensional transformation. The visual effect of channel error on the encoded pictures and its effect on the performance of the system measured in terms of the signal-to-noise ratio of the encoder using a binary symmetric channel is examined. The system shows little degradation at small or moderate levels of channel noise, however its performance is degraded significantly at high levels of channel error. The system is particularly attractive in the sense that the principle can be expanded to utilize interframe coding of television signals. Such a coding system would start by taking a two-dimensional transformation of each frame of the television signal, then it would encode the transformed signal in the temporal direction by a number of parallel DPCM encoders, thus exploiting the correlation of data in temporal as well as spatial directions.

The second system proposed uses a two-dimensional unitary transformation on the pictorial data divided into small blocks. The elements of each block in the transformed domain are ordered in a one-dimensional array and are coded by a bank of DPCM systems. A small block size reduces the number of arithmetic operations needed to obtain the transformed data, but it reduces the effeciency of the transformed system since the elements of various blocks remain correlated in the transformed domain. However, DPCM coding of the elements in the transformed domain take advantage of this correlation and improves the performance of the system.

II. One-dimensional Transformation and DPCM Coding

In the system proposed here the pictorial data is scanned to form N lines with appropriate vertical resolution, then each line is sampled at a Nyquist rate. The sampled image is then divided into arrays of M by N picture elements u(x,y), where x and y index the rows and the columns in each individual array such that the number of samples in a line of image is an integer multiple of M. One-dimensional unitary transformation of the data and its inverse are modeled by the set of equations

$$u_i(y) = \sum_{x=1}^{M} u(x,y)\Phi_i(x) \qquad \begin{matrix} i=1,2,\ldots,M \\ y=1,2,\ldots,N \end{matrix} \tag{1}$$

$$u(x,y) = \sum_{i=1}^{M} u_i(y)\Phi_i(x) \tag{2}$$

where $\Phi_i(x)$ is a set of M orthonormal basis vectors. The correlation of the

transformed samples $u_i(y)$ and $u_i(y+\tau)$ is given by

$$C_i(\tau) = \sum_{x=1}^{M} \sum_{\hat{x}=1}^{M} R(x,\hat{x},y+\tau)\Phi_i(x)\Phi_i(\hat{x}) \tag{3}$$

where $R(x,\hat{x},y,\hat{y})$ is the spatial autocovariance of the data.

Note that this equation indicates (1) the correlation of samples in each column of the transformed array is directly proportional to the correlation of sampled image in vertical direction, (2) the correlation of samples in various columns of the transformed array is different. Thus, a number of different DPCM systems should be used to encode each column of the transformed data. The block diagram of the proposed system is shown in Figure 1. A replica of the original image $u^*(x,y)$ is formed by inverse transforming the coded samples, i.e.,

$$u^*(x,y) = \sum_{i=1}^{n} v_i(y)\Phi_i(x), \quad n \leq M \tag{4}$$

The mean square value of coding error is

$$\varepsilon^2 = E \left\{ \frac{1}{MN} \sum_{y=1}^{N} \sum_{x=1}^{M} [u(x,y)-u^*(x,y)]^2 \right\} \tag{5}$$

Using equations (1) to (4) and assuming that $q_i(y)$, the quantization error encountered in the i^{th} DPCM system, is uncorrelated with $u_i(y)$ the coding error, ε^2 is

$$\varepsilon^2 = R(0,0,0,0) - \sum_{i=1}^{n} C_i(0) + E \left\{ \frac{1}{MN} \sum_{y=1}^{N} \sum_{x=1}^{n} [u_i(y)-v_i(y)]^2 \right\} \qquad (6)$$

where the first two terms are introduced because of using n (rather than M) DPCM systems. Study of DPCM systems has shown that [1]

$$E\ [u_i(y)-v_i(y)]^2 = E\ q_i^2(y) = K(m_i)e_i^2 \qquad (7)$$

where e_i^2 is the variance of the differential signal in the i^{th} DPCM and $K(m_i)$ is the quantization error of a variate with a unity variance in a quantizer with $(2)^{m_i}$ levels.

Analysis of Lloyd-Max quantizer has shown that $K(m_i)$ can be approximated farily accurately by

$$K(m_i) \cong b\ e^{-a\ m_i} \qquad (8)$$

where the best fit for a Gaussian variate is obtained using a = 0.5 ℓn 10 and b = 1.0 [4,5,7]. Study of other quantization techniques has indicated similar results for various probability density functions [8]. Figure 2 shows the functional form of $K(m_i)$ and the accuracy of approximation (8) using a = 0.5 ℓn 10, b = 2.0 for a random variable with a double-sided exponential probability density function using an instaneous companding quantizer [9].

From published results [1], [6] the variance of the differential signal in a DPCM system with the m^{th} order linear predictor is

$$e_i^2 = C_i(0) - \sum_{j=1}^{m} A_{ij}C_i(j) \qquad (9)$$

where A_{ij} are related to $C_i(j)$ by m algebraic equations.

Substituting (7) and (8) in (6), ε^2 is

$$\varepsilon^2 = R(0,0,0,0) - \sum_{i=1}^{n} C_i(0) + \frac{b}{M} \sum_{i=1}^{n} e^{-am_i} e_i^2 \tag{10}$$

where the error is defined in terms of n and m_i, i=1,...,n. Treating m_i as continuous variables and minimizing ε^2 with a constraint $\sum_{i=1}^{n} m_i = M_b$ will give

$$m_i = \frac{M_b}{n} + \frac{1}{a} \ln e_i^2 - \frac{1}{n} \sum_{i=1}^{n} \ln e_i^2 \tag{11}$$

where n is chosen such that ε^2 is minimum* and the quantizer in the i^{th} DPCM system will have $(2)^{m_i}$ levels. m_i as obtained from equation (11) is modified as discussed in [4] and [5]. Note that expressions (10) and (11) are similar to those obtained for coding the transformed data by memoryless quantizers, the difference being that here the variance of the differential signal rather than the variance of the transformed data is used.

Figure 3 shows the theoretical value of the coding error as given by (1) in terms of the peak-to-peak signal to rms noise ratio for a discrete random field with an autocorvariance

$$R(x,\hat{x},y,\hat{y}) = e^{-\alpha|x-\hat{x}|-\beta|y-\hat{y}|} \tag{12}$$

using Karhunen-Loeve, Hadamard, discrete Fourier, discrete Cosine [11], and Slant [12] transformations for $\alpha = 0.0545$, $\beta = 0.128$. These curves are obtained using a bank of n DPCM encoders with one-element predictors. The quantizers in the DPCM systems are of instaneous-companding type and are designed for the

*Ready and Wintz minimize ε^2 with respect to both n and m_i. It gives the same end result requiring less computations [10].

probability density functions of the differential signals in the DPCM systems. Experimental results (see Figure 4) indicate that the probability density functions of the differential signals are double-sided exponential functions.

Figure 5 shows that increasing the block size M improves the theoretical performance of the proposed one-dimensional systems, however the improvement becomes negligible for values of M larger than 8. This makes this coding system less sensitive to the block size than the standard transform coding systems which use memoryless quantizers.

III. Two-Dimensional Transformation and DPCM Coding

The most severe limitation of the two-dimensional transform coding technique is the large number of operations needed for coding pictorial data and the resulting equipment complexity. This limitation becomes less significant when the picture is divided into smaller block sizes then encoded, but this also limits the efficiency of the system since the elements of various blocks remain correlated in the transformed domain. A coding system utilizing two-dimensional transformation and a bank of DPCM systems would exploit this correlation, thus improving the coding efficiency of the system.

In this system the sampled image $u(.,.)$ of N by N elements is divided into arrays of M by M elements. The location of each array in the image is indexed by subscripts (k,ℓ). A two-dimensional unitary transformation of each block of image is obtained and is ordered to form a one-dimensional array U_i, $i = 1,2,\ldots, M^2$ where $U_i(k,\ell)$ refers the the i^{th} member of this array which is obtained by transforming the $(k,\ell)^{th}$ block of the data. This transformation and its inverse are modeled as

$$U_i(k,\ell) = \sum_{y=1}^{M} \sum_{x=1}^{M} u(kM+x,\ell M+y)\Phi_i(x,y) \tag{13}$$

$$u(kM+x,\ell M+y) = \sum_{i=1}^{M^2} U_i(k,\ell)\Phi_i(x,y) \tag{14}$$

where $\Phi_i(x,y)$ are a set of orthonormal basis matrices. The elements of $U_i(k,\ell)$ arrays for various values of i, $(i=1,2,\ldots,M^2)$ are correlated, thus could be coded by M^2 DPCM systems. The analysis of this problem is similar to one discussed in Section II. The coding error and the bit assignment rules are

$$\varepsilon^2 = R(0,0) - \sum_{i=1}^{n} C_i(0) + \frac{b}{M} \sum_{i=1}^{n} e^{-am_i} e_i^2 \tag{15}$$

$$m_i = \frac{M_b}{n} + \frac{1}{a} \ell n\, e_i^2 - \frac{1}{n} \sum_{i=1}^{n} \ell n\, e_i^2 \tag{16}$$

where

$$C_i(\tau) = \sum_{\hat{y}=1}^{M} \sum_{y=1}^{M} \sum_{x=1}^{M} \sum_{\hat{x}=1}^{M} R(x-M\,\tau-\hat{x},y-\hat{y})\Phi_i(x.y)\Phi_i(\hat{x},\hat{y}) \tag{17}$$

and e_i^2 is the variance of the differential signal in i^{th} DPCM system.

$$e_i^2 = C_i(0) - \sum_{j=1}^{m} A_{ij}C_i(j) \tag{18}$$

Note that in (16), R(.,.) is the autocovariance of the image u(.,.) and it is assumed that u(.,.) is a covariance stationary random field.

The last term in (15) indicates the improvements due to the use of DPCM systems. Replacing the DPCM systems by memoryless quantizers will give a similar expression for ε^2 where e_i^2 is replaced by $C_i(0)$. Thus the coding improvement due to the use of DPCM encoders depends on the relative values of e_i^2 and

$C_i(0)$ which in turn depends on R(.,.) and the block size M. In addition, note that the coding error of the two-dimensional transform encoders are obtained from (15) using $C_i(0)$ for σ_i^2 and b = 1, since it is generally assumed that the transformed data possesses a normal distribution and according to (8), b = 1. This assumption is not accurate for small block sizes and this partially accounts for the difference between the experimental and the theoretical performance of the standard two-dimensional transform encoders [4]. In the hybrid systems ϵ^2 is evaluated for b = 2 which gives a larger theoretical value for the coding error, however experimental results indicate that the differential signal in the hybrid systems posses an exponential density regardless of the block size (see Figure 4). This results in a better agreement between the theoretical and the experimental results in the hybrid system.

Figure 6 shows the performance of the two-dimensional system for the random field that was considered in Section II. Here a block size of 4 by 4 is used, the bit assignment is according to (16), and the DPCM encoders are identical to those employed for the one-dimensional system. Figure 7 shows the performance of the two-dimensional systems for various block sizes M. The results are similar to those observed for the one-dimensional system.

IV. Experimental Results and the Noisy Channel

The coding systems discussed in Sections II and III were simulated on a digital computer and were used to encode the picture shown in Figure 9a. This picture is composed of 256 by 256 picture elements, each element quantized to 64 levels. Values of α and β for (12) were approximated from Figure 9a and are 0.0545 and 0.128 respectively. The experimental results for the hybrid system

using one-dimensianal transformations is obtain using a block size of 16 and for the hybrid system using two-dimensional transformations for a block size of 4 by 4. The results of the proposed systems for the cascade of one-dimensional Hadamard, Fourier, and Karhunen-Loeve transformations with the DPCM systems as well as the cascade of two-dimensional Hadamard Transformation and the DPCM systems are plotted in Figure 8. This figure also shows the performance of a simple third-order DPCM system discussed in [6] and the performance of the encoder using two-dimensional Hadamard transform and a block quantizer discussed in [4], which is included here for comparison. The two-dimensional Hadamard system uses a block size of 16 by 16 and gives a better signal-to-noise ratio than the two-dimensional hybrid system at low bit rates, however the two-dimensional hybrid system performs better at high bit rates. Note that the two-dimensional hybrid encoder uses a block size of 4 by 4 and its performance would have been inferior to the performance of the two-dimensional transform encoder at all bit rates if it were not a hybrid system. The one-dimensional hybrid systems perform better than the two-dimensional and also the simple DPCM system. The performance of the two-dimensional hybrid system is similar to the performance of the simple two-dimensional system in the sense that it performs better than the DPCM system at low bit rate, however it is worse at a higher bit rate. The encoded pictures corresponding to the one-dimensional and the two-dimensional hybrid systems as well as the system using two-dimensional Hadamard transform are shown in Figures 9, 10, and 11 for one and two bits per pixel. Good coding results are obtained at a bit rate of 2 bits/pixel where the pictures encoded at one bit/pixel are acceptable; though some degradation is noticeable. We also note that the pictures encoded by the hybrid systems are less degraded than

their counterparts encoded by the two-dimensional transform coder.

To study the effect of channel error in the performance of the hybrid encoders a binary symmetric channel was simulated on a digital computer to link the transmitter with the receiver. This channel would operate on each binary digit independently changing each digit from 0 to 1 or from 1 to 0 with probability p and leaving the digit unchanged with probability 1-p. Then the same receiver discussed in Section II is used to reconstruct the encoded picture from the string of binary digits at the output of the channel. Figure 12 shows the degradation of the one-dimensional hybrid system using Hadamard transformation and the DPCM encoders, in terms of signal-to-noise ratio, for bit-error probabilities of $p = 10^{-2}$, $p = 10^{-3}$, and $p = 10^{-4}$. It is observed that the performance of the hybrid system is affected very little for a channel noise corresponding to a bit-error probability of 10^{-4}, however it increases significantly at bit-error probabilities of 10^{-3} and 10^{-2}. Also, it is noted that the effect of channel error is more emphasized at higher bit rates, particularly for bit-error probabilities of 10^{-3} and 10^{-2}. Figure 13 shows the encoded pictures using the above hybrid system at bit rates of one and two bits per pixel for bit-error probabilities of 10^{-2}, 10^{-3}, and 10^{-4}. These figures show that aside from the exceptionally high bit-error probability of 10^{-2}, the degradation of the encoded pictures because of noisy channel is not significant. Visible degradations are in the form of short runs of 16 pixels (the block length of one-dimensional transformations) which is due to the error in the output of the DPCM system that encodes the lower components of the transformed signal. We note that there is no propagation of the error in the vertical direction as is commonly observed in simple DPCM systems.

The hybrid system we have investigated is one that uses the optimum coefficients A_1, A_2,...,A_n in the predictors employed in both the receiver and the transmitter of the system (see Figure 1). These coefficients depend upon the statistics of the particular picture that we are encoding. To make the encoder independent of the signal statistics, a common value of unity can be used for A_1 through A_n. This makes the bank of DPCM systems similar to the differential encoders that have been used in practice [3]. Figure 14 shows the performance of the hybrid system discussed above using a common value of unity for A_i, i=1,...,n at various bit rates and bit-error probabilities. This figure also shows the performance of the same encoder using optimum coefficients and a noiseless channel. The degradation due to making these coefficients unity is about one dB and it is almost the same at various bit rates. The behavior of this system for various bit-error probabilities is very similar to the performance of the encoder using optimum coefficients shown in Figure 12.

Encoded pictures using this suboptimum encoder for one and two bits per pixel at various bit-error probabilities are shown in Figure 15. Comparison of Figures 13 and 15 shows that there is no noticeable difference in the performance of the optimum and suboptimum encoder.

To compare the performance of the hybrid system with the encoder using two-dimensional transform and block quantization for a noisy channel, the binary symmertric channel discussed aboveis simulated in the system that uses two-dimensional Hadamard transform and a block quantizer for a block size of 16 by 16. Figure 16 shows the performance of this system in terms of signal-to-noise ratio at various bit rates for bit-error probabilities of 10^{-4}, 10^{-3}, and 10^{-2}. Two pictures encoded by this system using one and two bits per pixel and bit-error

probability of 10^{-3} are shown in Figure 17. Comparison of these pictures with those in Figures 11 and 13 indicates that the subjective quality of the resultant pictures is affected more because of channel noise when coded by the hybrid system then it is using the two-dimensional transform technique, though the hybrid system produces pictures with a better overall quality.

V. Conclusions

The hybrid system using the two-dimensional transformations and the DPCM encoders is essentially the standard two-dimensional transform coding system where the encoding efficiency is improved by exploiting the inter-block correlation in the data. The system is attractive since a small block size could be used reducing the computational complexity without affecting the performance of the system.

The hybrid system that uses one-dimensional transformation and DPCM systems attempts to uncorrelate the data in one spatial direction by the transform technique and in the other spatial direction by the DPCM encoders. The quantization and the channel error are introduced in the transform domain, where it is less objectionable to a human viewer. The fact that the system uses a one-dimensional transformation, it is not sensitive to the block size, and allows parallel operation on the data, makes the system attractive from an implementational viewpoint. The significant result is the improved performance of the system as indicated by both theoretical and experimental results that surpasses the performance of both the DPCM and the two-dimensional transform coding systems and the fact that the performance of the encoder is not impaired significantly for small or moderate levels of channel noise.

Acknowledgement

The author wishes to thank Dr. Wen-hsiung Chen of the University of Southern California for his efforts in simulating the noisy channel in the system.

References

1. J. B. O'Neal, Jr., "Delta modulation quantizing noise - Analytical and computer simulation results for Gaussian and television input signals," Bell Syst. Tech. J., Vol. 45, pp. 117-142, Jan. 1966.

2. W. K. Pratt, J. Kane and H. C. Andrews, "Hadamard transform image coding," Proc. IEEE, Vol. 57, pp. 58-68, Jan. 1969.

3. J. O. Limb, C. B. Rubinstein and K. A. Walsh, "Digital coding of color picturephone signals by element-differential quantization," IEEE Trans. Commun. Tech., Vol. COM-19, No. 6, pp. 992-1005, Dec. 1971.

4. A. Habibi and P. A. Wintz, "Image coding by linear transformations and block quantization," IEEE Trans. Commun. Technol., Vol. COM-19, pp. 50-63, Feb. 1971.

5. J. J. Y. Huang and P. M. Schultheiss, "Block quantization of correlated Gaussian random variables," IEEE Trans. Commun. Syst., Vol. CS-11, pp. 289-296, Sept. 1963.

6. A. Habibi, "Comparison of nth-order DPCM encoder with linear transformations and block quantization techniques," IEEE Trans. Commun. Tech., Vol. COM-19, No. 6, pp. 948-956, Dec. 1971.

7. P. A. Wintz and A. J. Kurtenback, "Waveform error control in PCM telemetry," IEEE Trans. Inform. Theory, Vol. IT-14, pp. 650-661, Sept. 1968.

8. A. Habibi, "Performance of zero-memory quantizers using rate distortion criteria," to be published.

9. B. Smith, "Instaneous companding of quantized signals," Bell Syst. Tech. J., Vol. 36, pp. 44-48, Jan. 1951.

10. P. J. Ready and P. A. Wintz, "Multispectral data compression through transform coding and block quantization," School of Elec. Engr., Purdue University, Lafayette, Ind., Tech. Rep. TR-EE 72-29, May 1972.

11. N. Ahmed, T. Natrajan and K. R. Rao, "Discrete cosine transform," IEEE Trans. on Computers, Vol. C-00, No. 0,pp. 000-000, 1973.

12. W. K. Pratt, L. R. Welch and W. Chen, "Slant transforms for image coding," Proceedings of the Symposium on Application of Walsh Functions, March 1972.

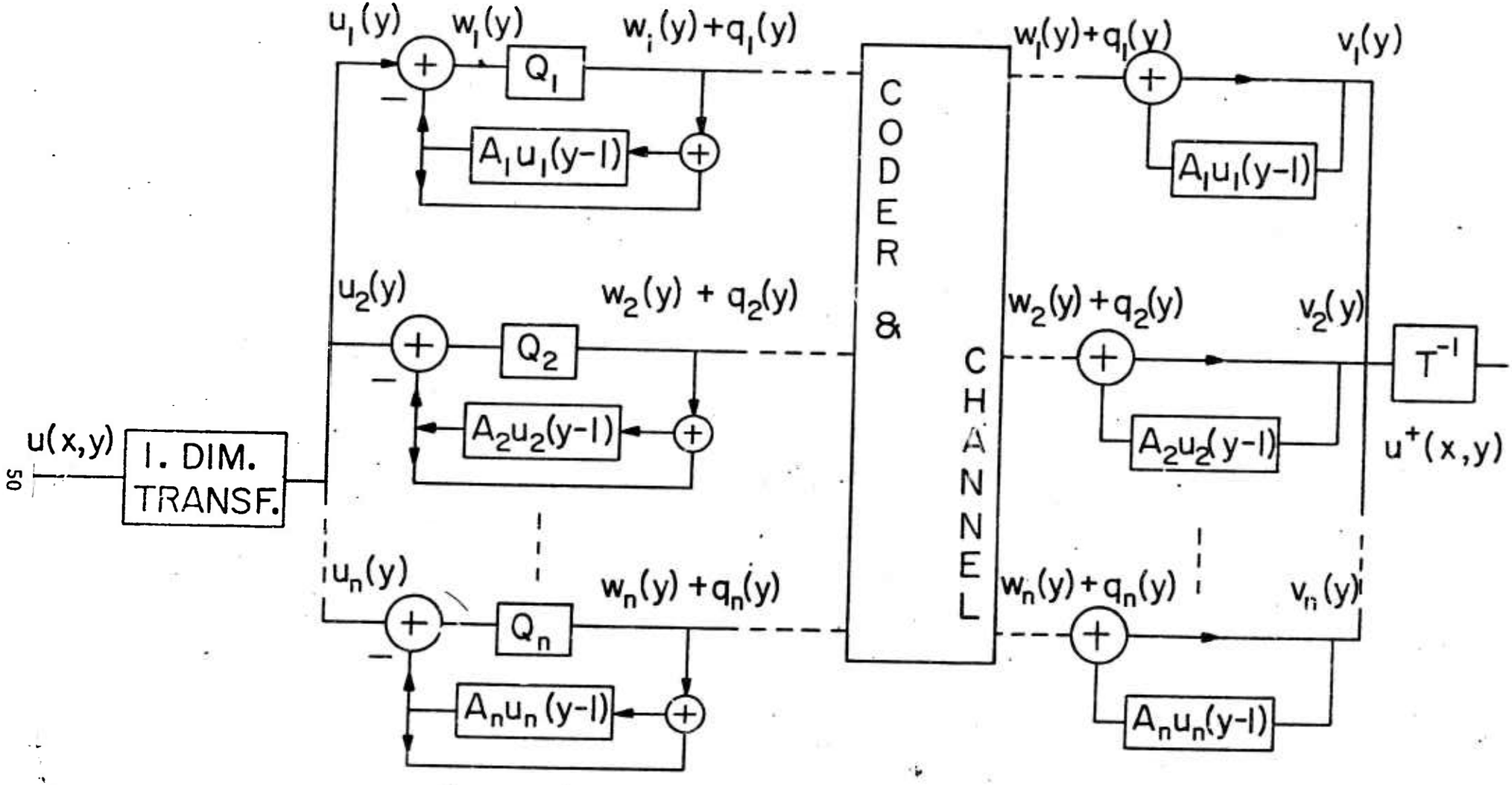


Fig. 1. Block diagram of the hybrid system using a cascade of one-dimensional transformations and a bank of DPCM systems to encode pictorial data.

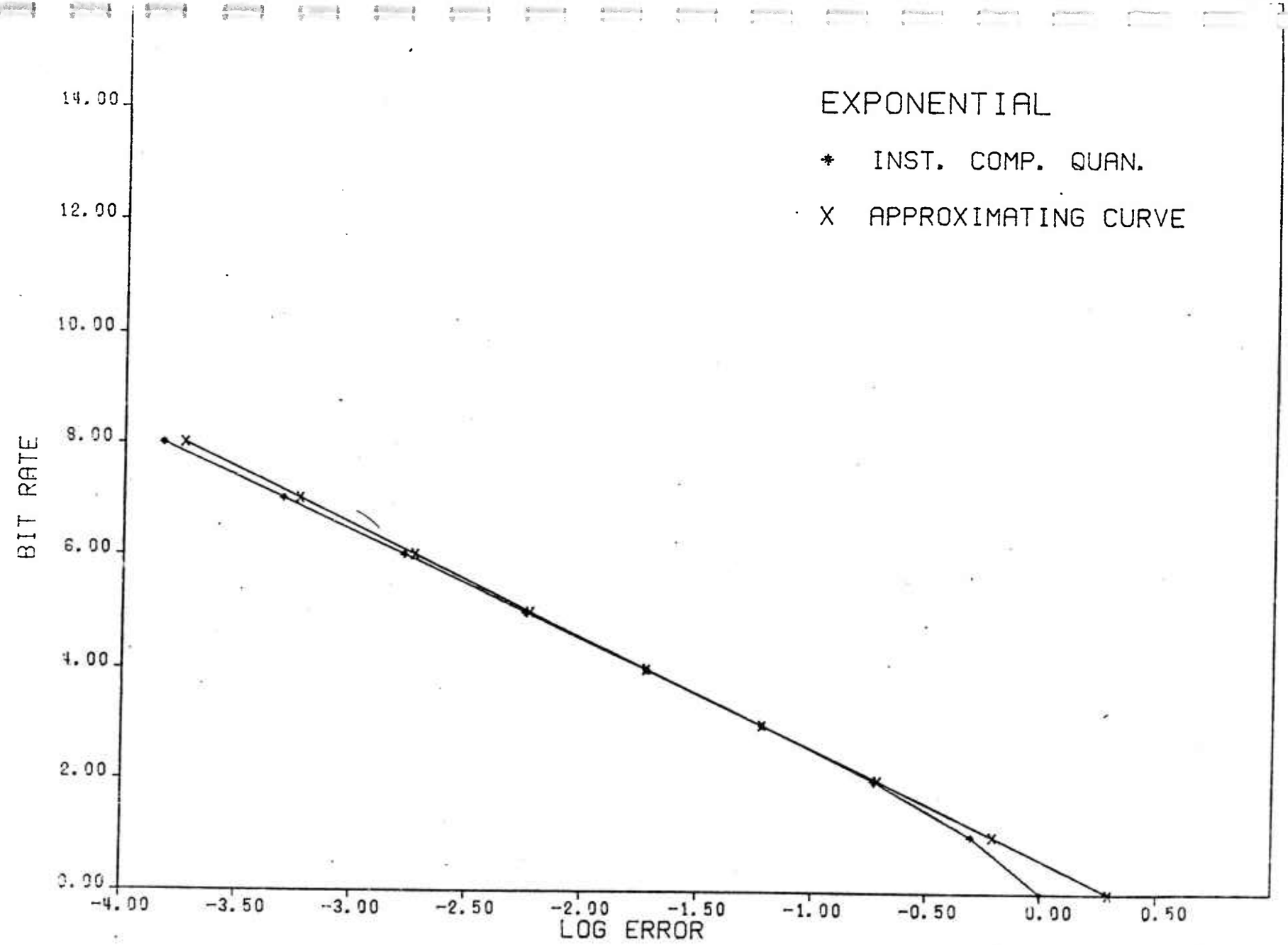


Fig. 2 Bit rate versus the logarithm of the quantization error and its linear approximation for a variate with a double sided exponential probability density function using an instaneous-companding quantizer.

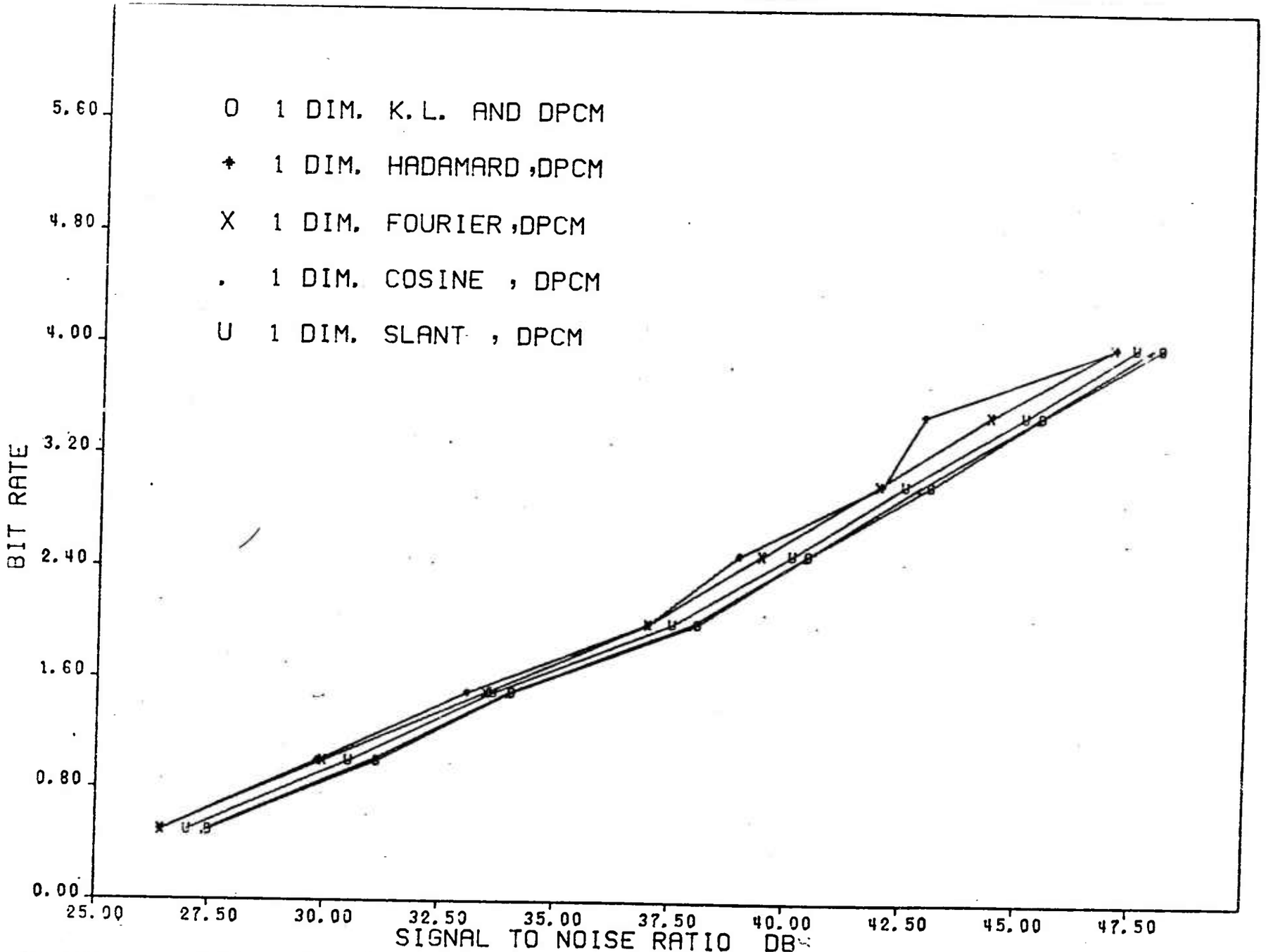


Fig. 3. Bit-rate versus the signal-to-noise ratio for the proposed one-dimensional hybrid systems for the discrete random field.

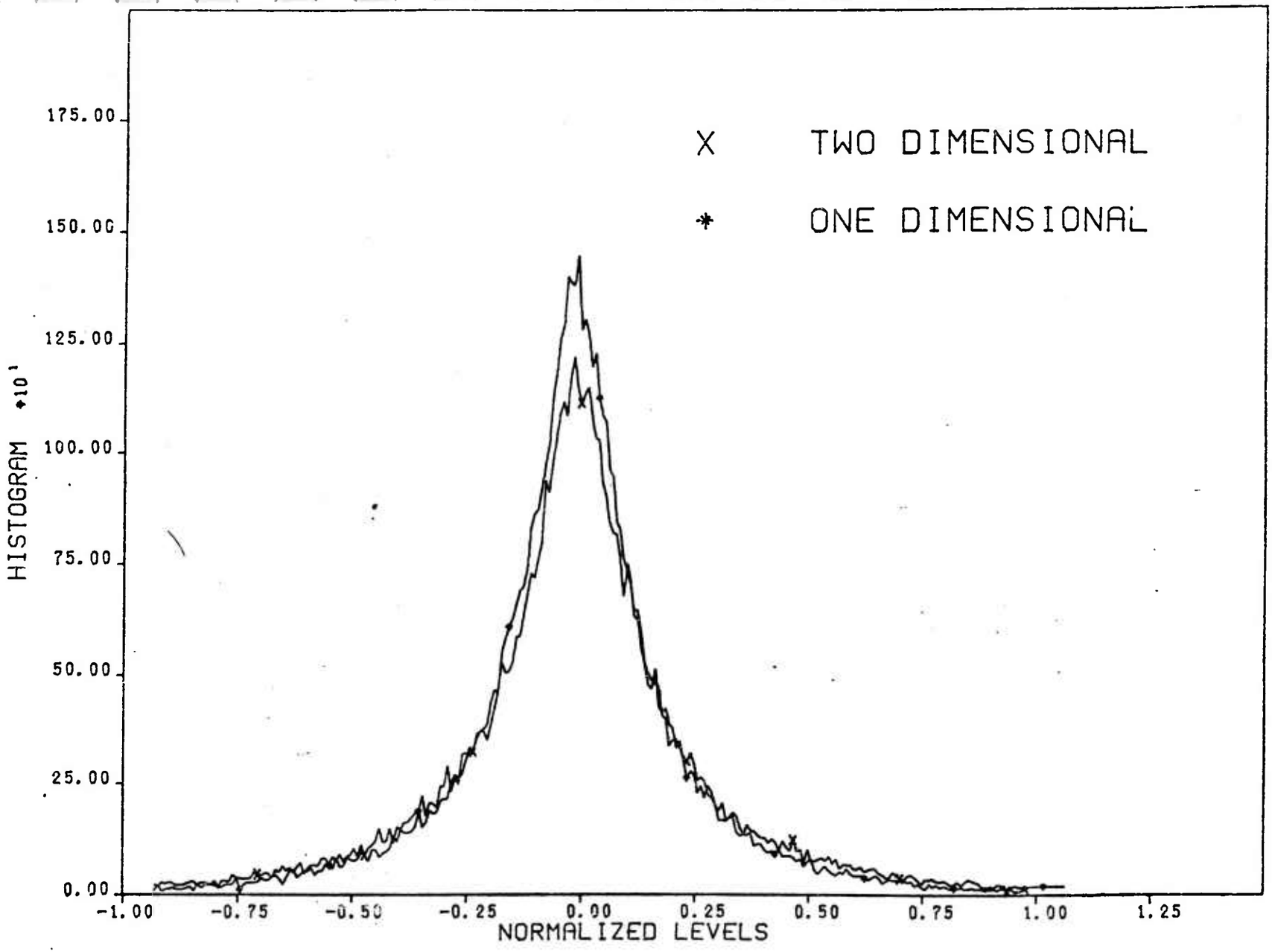


Fig. 4 Histogram of the differential signals in the DPCM systems for one-dimensional (M=16) and two-dimensional (4 X 4) Hadamard transformations.

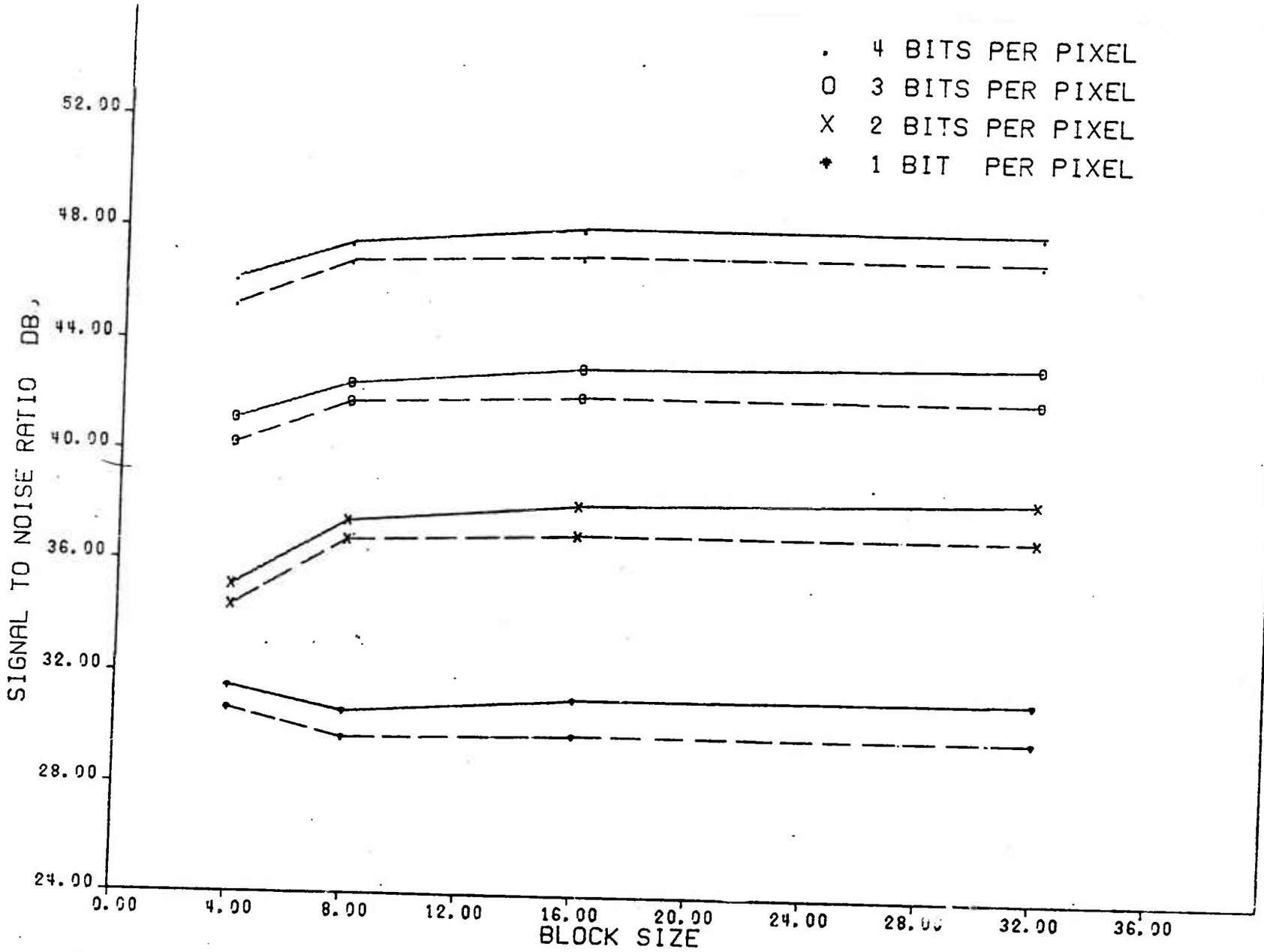


Fig. 5 Theoretical performance of the proposed one-dimensional encoders versus block size M. The solid and the dashed lines refer to the Karhunen-Loeve and the Hadamard transforms, respectively.

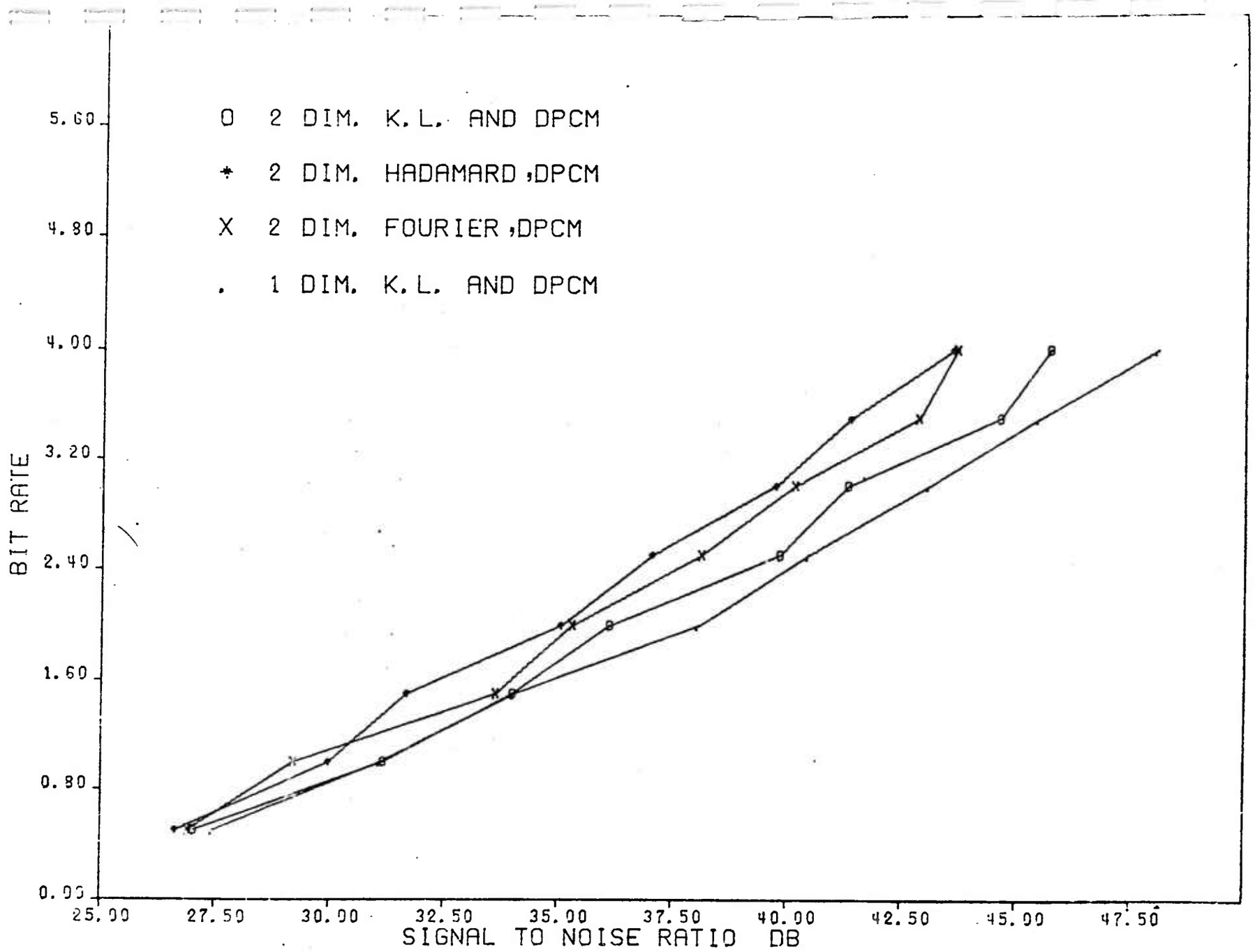


Fig. 6. Bit rate versus the signal-to-noise ratio for the proposed two-dimensional hybrid systems for the discrete random field. Performance of the hybrid system using one-dimensional K.L. transform and DPCM is included for comparison.

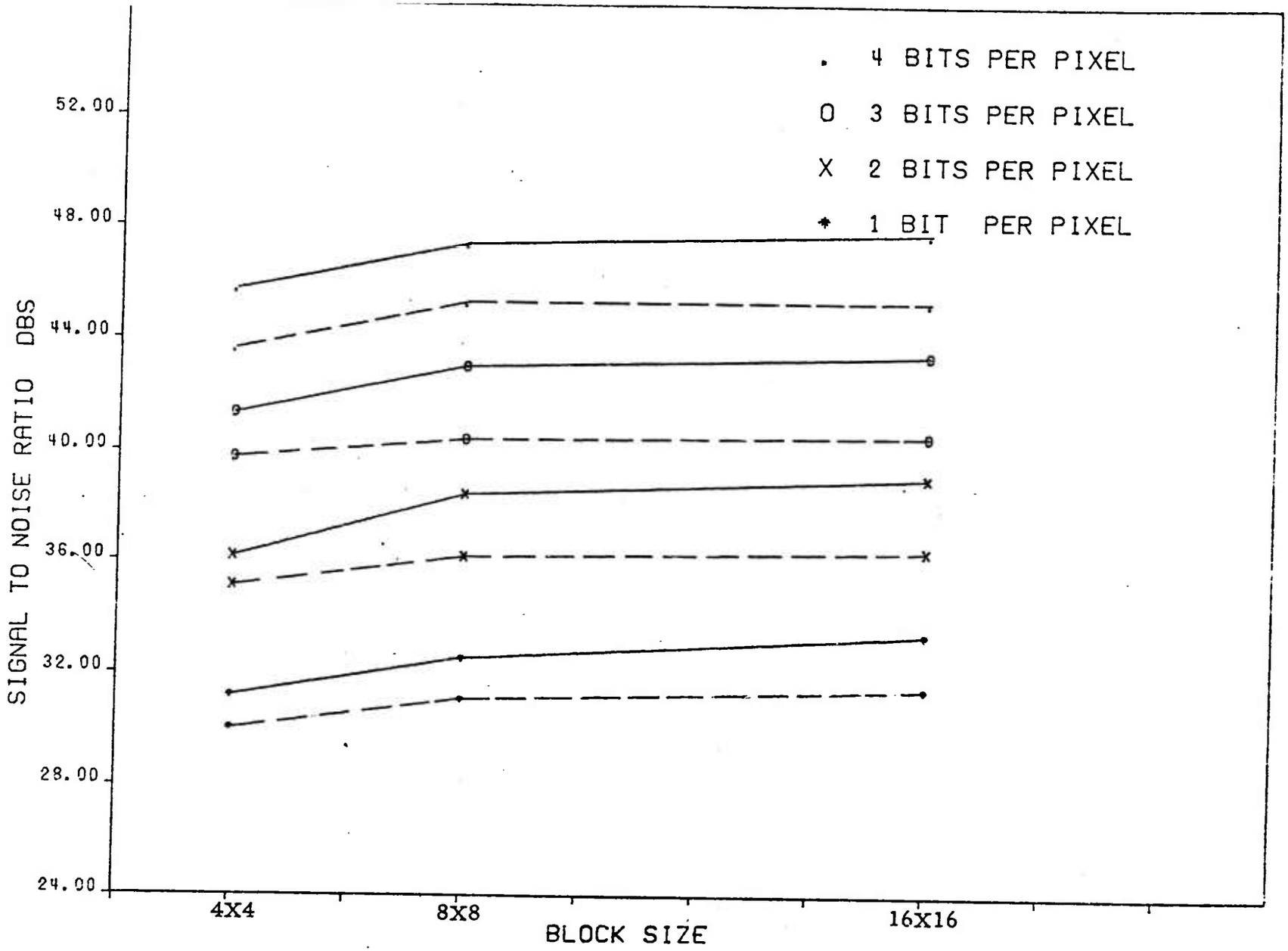


Fig. 7 Theoretical performance of the proposed two-dimensional encoders versus block size M X M. The solid and the dashed lines refer to the Karhunen-Loeve and the Hadamard transforms, respectively.

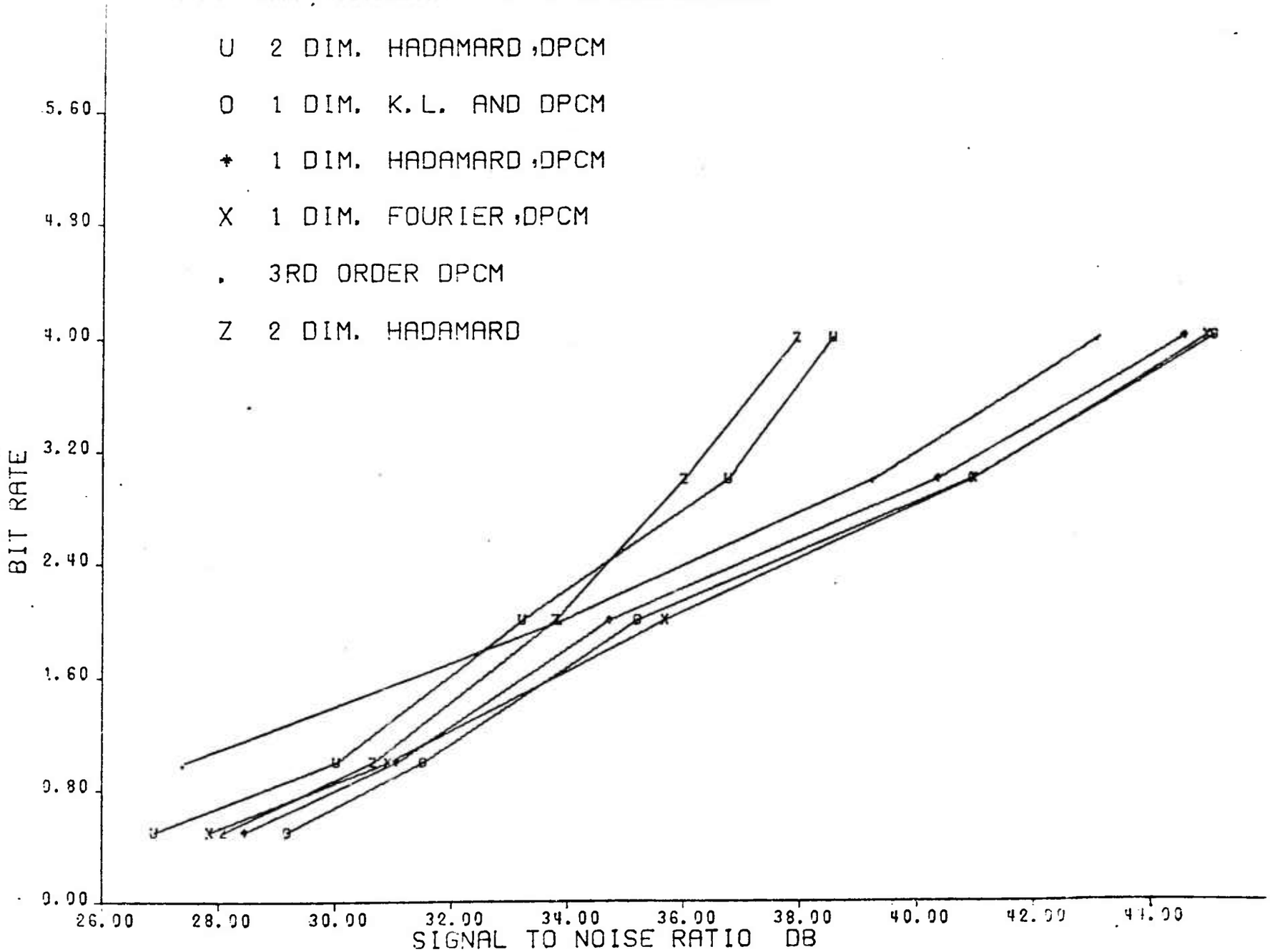


Fig. 8. Experimental results comparing the performance of the proposed hybrid systems using the picture in Fig. 9(a). The performance of a third-order simple DPCM and the encoder using two-dimensional Hadamard transform and a block quantizer is included for comparison.

(a) Original

(b) Karhunen-Loeve

(c) Fourier

(d) Hadamard

Figure 9. The original and the encoded pictures using one-dimensional transformations and DPCM systems, 1 bit/pixel, M = 16.

(a) Karhunen-Loeve

(b) Fourier

(c) Hadamard

Figure 10. The encoded pictures using the cascade of one-dimensional transformations and DPCM systems, 2 bits/pixel, M = 16.

(a) 1 bit/Pixel
Two-Dimensional Hybrid

(b) 2 bits/Pixel
Two-Dimensional Hybrid

(c) 1 bit/Pixel
Two-Dimensional Hadamard

(d) 2 bits/Pixel
Two-Dimensional Hadamard

Figure 11. (a) and (b) are coded by the hybrid system using two-dimensional Hadamard transform (4 by 4) and DPCM. (c) and (d) are coded by the system using two-dimensional Hadamard transform (16 by 16) and a block quantizer.

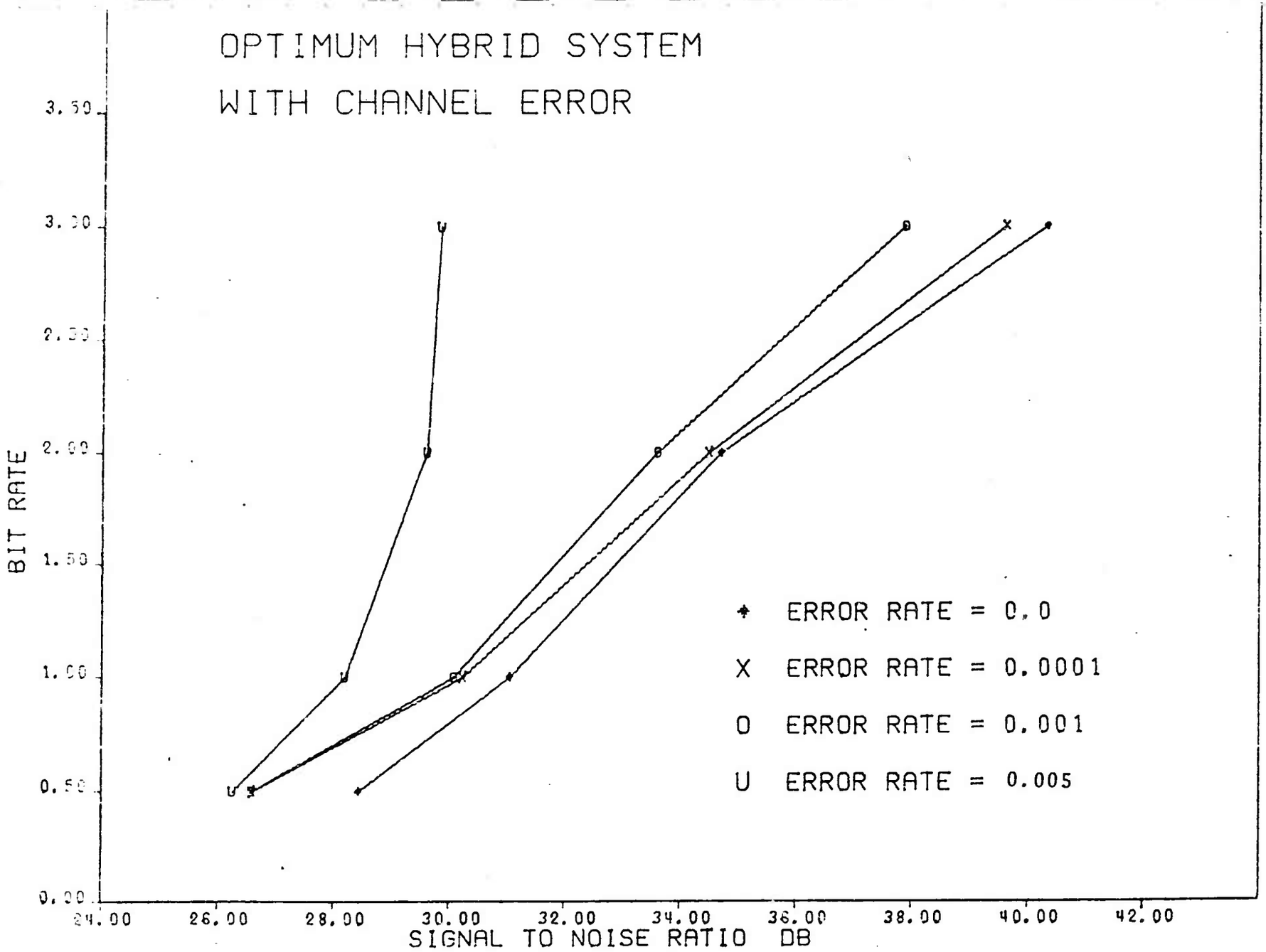


Fig. 12. Performance (experimental results) of optimum hybrid system (A_i's have correct value) using one-dimensional Hadamard transform and DPCM encoders for a noisy channel.

(a) 1 bit, $P=10^{-4}$

(d) 2 bits, $P=10^{-4}$

(b) 1 bit, $P=10^{-3}$

(e) 2 bits, $P=10^{-3}$

(c) 1 bit, $P=10^{-2}$

(f) 2 bits, $P=10^{-2}$

Figure 13. Coded by the hybrid system using one-dimensional Hadamard transform and DPCM encoders employing a noisy channel.

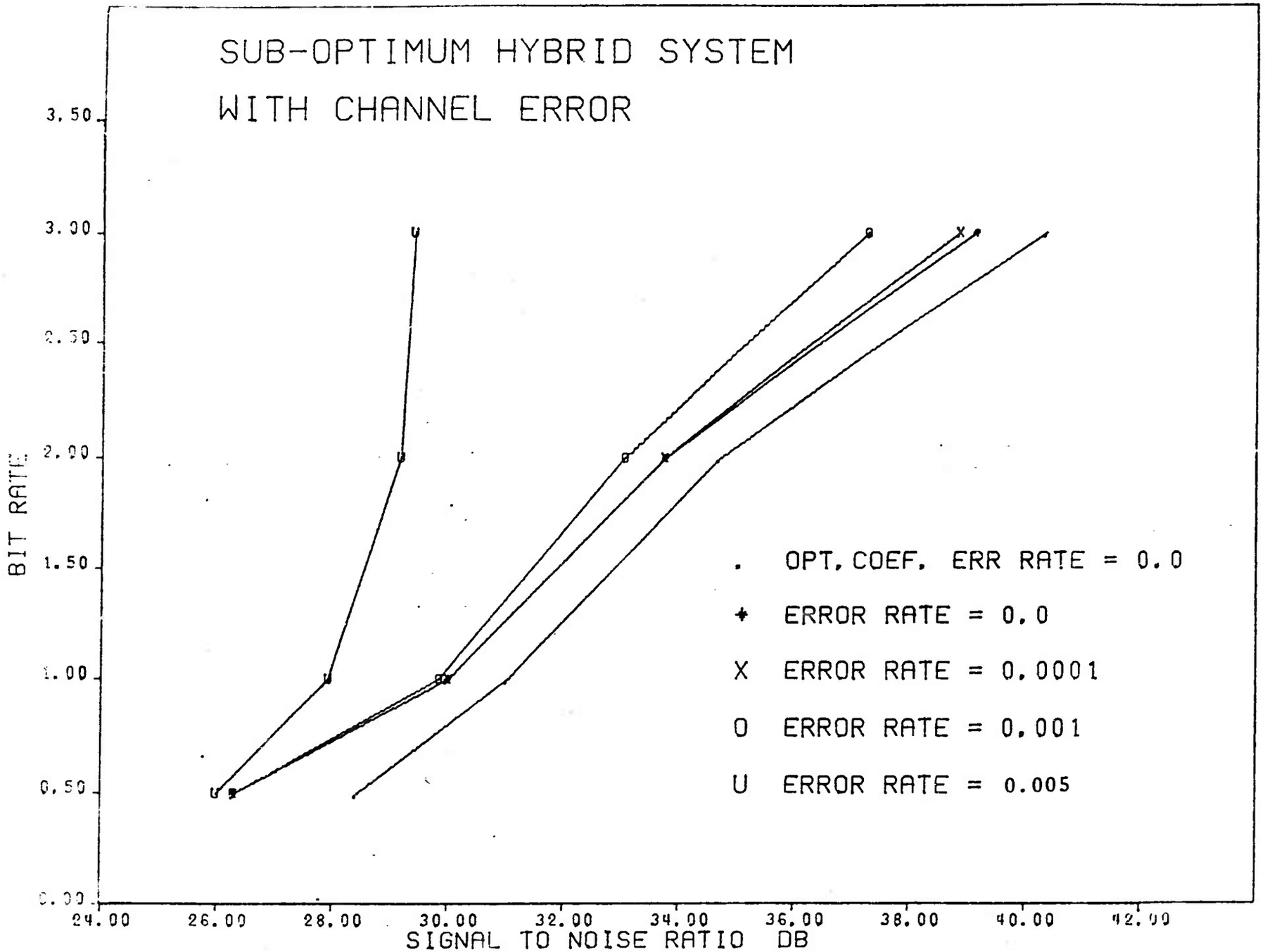


Fig. 14. Performance (experimental results) of suboptimum hybrid system ($A_j = 1$) using one-dimensional Hadamard transform and DPCM encoder for a noisy channel. Performance of optimum hybrid encoder is included for comparison.

(a) 1 bit, P= 10^{-4}

(d) 2 bits, P= 10^{-4}

(b) 1 bit, P=10^{-3}

(e) 2 bits, P=10^{-3}

(c) 1 bit, P=10^{-2}

(f) 2 bits, P=10^{-2}

Figure 15. Coded by the hybrid system using one-dimensional Hadamard transform and simplified DPCM encoders (A_i=1) employing a noisy channel.

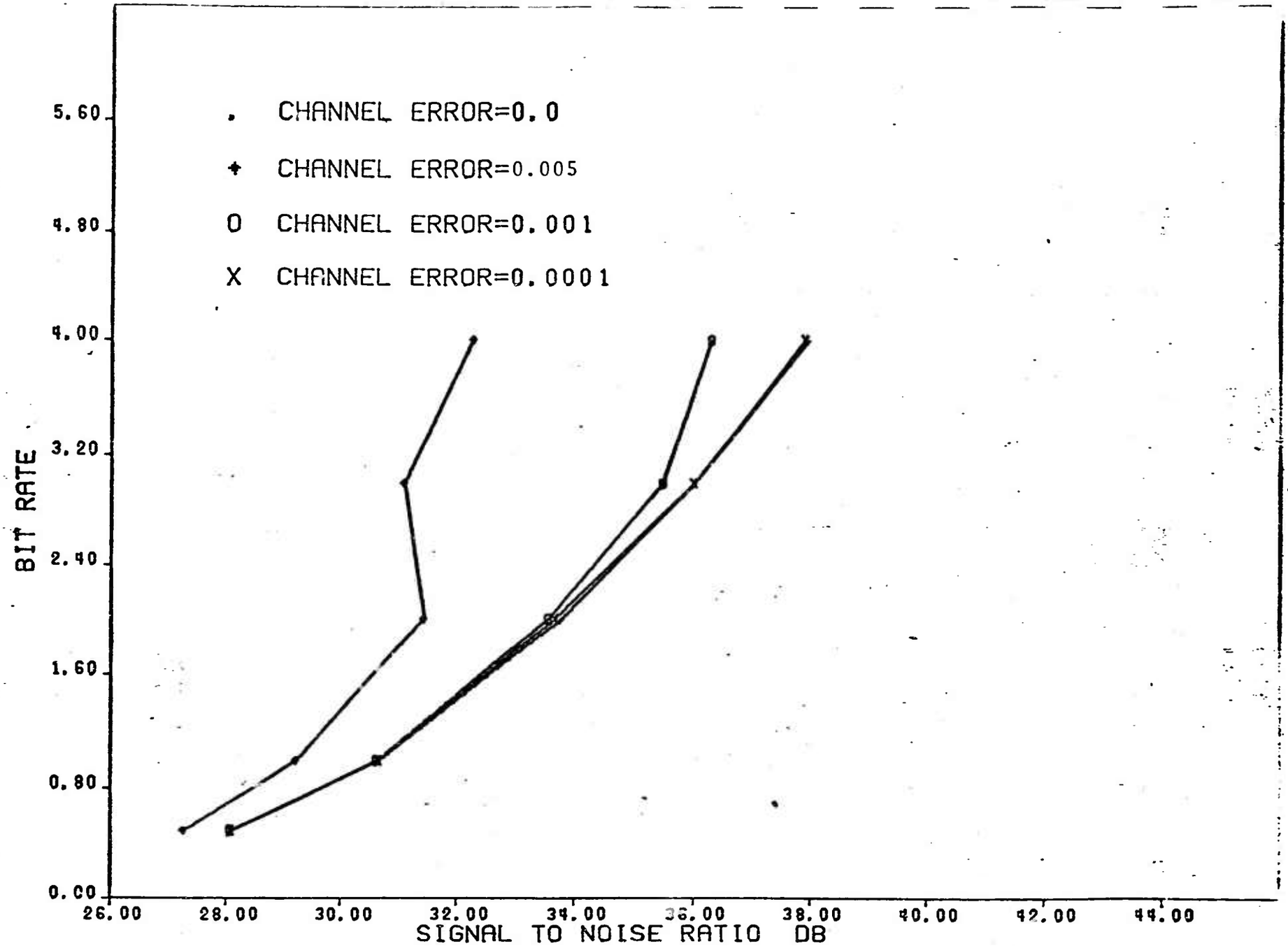


Figure 16. Experimental results showing the encoder that uses two-dimensional Hadamard transform (16x16) and a block quantizer for a noisy channel.

(a) 1 bit/pixel , $P=10^{-2}$

(b) 2bits/pixel , $P=10^{-2}$

Figure 17. Coded by the system using two-dimensional Hadamard transform (16 by 16) and a block quantizer with a noisy channel.

APPENDIX C

REAL TIME DISCRETE FOURIER TRANSFORMS USING SAW DEVICES

REAL TIME DISCRETE FOURIER TRANSFORMS USING SURFACE ACOUSTIC WAVE DEVICES

J. M. Alsup, R. W. Means, and H. J. Whitehouse
Naval Undersea Center, San Diego, California

Abstract

A real time Discrete Fourier Transform (DFT) module is presented using surface acoustic wave devices. The chirp - z algorithm is used to compute the DFT. It requires premultiplication of the time signal by a discrete complex FM chirp. Since the algorithm is complex it is implemented in real arithmetic by four convolvers operating in parallel. A 63 tap transversal filter performs the convolution required for a length 32 DFT. Each transversal filter has uniformly spaced taps whose weights are the cosine or sine components of a complex chirp. The computation time of the device is proportional to the number of samples, N, in the time signal. If N is a power of 2 the computation time of the Fast Fourier Transform (FFT) is $N \log_2 N$. This device is thus a factor of $\log_2 N$ faster than a FFT using a multiplier of the same speed. It is also a low power, light weight device.

To Appear In Proc. IEE International Specialist Seminar on Component Performance and System Applications of Surface Acoustic Wave Devices, Aviemore, Scotland, U. K., Sept. 25-27, 1973.

REAL TIME DISCRETE FOURIER TRANSFORMS USING SURFACE ACOUSTIC WAVE DEVICES

J. M. Alsup, R. W. Means, and H. J. Whitehouse

Introduction

A transformation of considerable importance for signal processing is the discrete Fourier transform (DFT). It may be calculated using the chirp-Z transform (CZT) algorithm, [1,2] a method which is suitable for implementation via acoustic surface wave devices as well as other forms of the transversal filter. The CZT algorithm decomposes the DFT into a premultiplication by a discrete chirp, a convolution with a discrete chirp, and a postmultiplication by a discrete chirp. This is illustrated in Figure 1 and by the following equations for an N-point DFT:

$$G_m = \sum_{n=0}^{N-1} e^{-j2\pi mn/N} g_n;$$

but,

$$-2mn = -m^2 + (m-n)^2 - n^2;$$

therefore,

$$G_m = e^{-j\pi m^2/N} \sum_{n=0}^{N-1} e^{j\pi(m-n)^2/N} \left(e^{-j\pi n^2/N} g_n \right).$$

Since this algorithm requires the use of complex numbers, its implementation with real hardware requires the use of four convolvers. The real and imaginary components of the input g_n are multiplied by the real and imaginary components of the premultiplier-chirp and combined in pairs to drive the inputs of four chirp convolvers. The convolver outputs are combined in pairs and multiplied by the real and imaginary components of the postmultiplier-chirp and combined again to provide the real and imaginary components of the output G_m.

The computation time of the device is proportional to the number of samples, N, in the time signal. If N is a power of 2 the computation time of the Fast Fourier Transform (FFT) is $N \log_2 N$. This device is thus a factor of $\log_2 N$ faster than a FFT using a multiplier of the same speed. It is also a low power, light weight device.

Transversal Filter Convolver

The convolution operation may be implemented in highly parallel form as a transversal filter. Such a filter is diagrammed in Figure 2 along with the equivalent matrix multiplication. The filter consists of a delay line which propagates the input signal g_n along with discrete, equally-spaced taps which sample the signal along the delay line. The taps are weighted by the values b_n which correspond to the real and imaginary components of

The authors are with the Naval Undersea Center, San Diego, California 92132, USA.

the discrete chirp: $b_n = \cos(\pi n^2/N)$ or $\sin(\pi n^2/N)$. When N is even the tap weights repeat and circular convolution can be used. The output of the filter's delay line is shown as connectable to its input so as to be able to perform circular convolution. The complete DFT via the CZT algorithm is shown in Figure 3. Parallel implementation of the complex arithmetic is shown for simplicity.

Surface Wave CZT Convolver

For signal processing at sample rates above 5 MHz, the acoustic surface wave technology is a very suitable candidate for implementation of a CZT convolver. Not only can the sine and cosine tap weightings be easily implemented (by adjusting the finger overlap in the interdigital transducer taps), but all four convolvers required for the algorithm can be fabricated on a single substrate.

Figures 4 and 5 show a CZT mask used at the Naval Undersea Center to fabricate a surface wave 4-channel convolver. In this instance, circular convolution is simulated by using a repeating structure of 63 taps to implement a 32-point transform: the last 31 taps are a replication of the first 31 taps.

Features of the Design

The launch transducers, located at the extreme right and left ends of the mask, have an aperture which is common to two of the tapping structures. Thus, even though there are four receiving structures which constitute the four transversal filters, there are but two acoustic paths each of which is occupied by a sine and cosine pair of tapping structures. Acoustic waves may be launched from either end since the tapping structures are symmetric. Since the launch aperture is about 318 acoustic wavelengths wide and the maximum propagation distance is about 851 acoustic wavelengths long, it can be established that the farthest tap is well within the near field of the launch transducer and that the acoustic waves travel across the substrate in well-collimated beams. The fractional 'far-field length' involved is the ratio $851/(318)^2 \doteq .008$.

The mask is transferred to an ST-cut quartz substrate as a pattern of aluminum fingers oriented for acoustic propagation in the x-direction. The size of the crystal is 3.5 inches by 1.75 inches by (1/8) inch.

The launch transducers and each receive tap are composed of three pairs of interdigitated fingers spaced at .0044-inch intervals. The finger width and the finger gap are equal at .0011 inch. Since the velocity of propagation is 3158 m/s, the resulting center frequency is 28.3 MHz.

A close-up view of the tapping structure shows how the finger groups are weighted (Figure 5). The finger is cut short of the full possible aperture if a weight smaller than one is desired. Finger tips are retained to minimize possible acoustic diffraction due to differential mass loading across the aperture. A tap of weight zero is implemented by a space (no finger group).

The ratio of group spacing to wavelength was originally designed to be six, but artwork limitations resulted in a ratio of 5.91. Thus, the

inter-tap delay became 0.209 μsec and the corresponding chirp sample frequency, 4.78 MHz. The resulting fractional bandwidth is 4.78/2(28.3) = .084.

The rate of calculation of the DFT is thus determined: 32 points per 6.69 μsec. When calculating such a rate in the manner described, it is assumed that two units are multiplexed so that one is calculating while the other is loading with new data.

The ends of the quartz crystal were rounded so as to permit propagation of the acoustic waves around the ends of the back side of the crystal where they are absorbed by a thin layer of silastic room-temperature-vinyl (RTV) distributed over the back surface.

The crystal was mounted in a milled aluminum box with separate compartments in the underside to accommodate electrical matching circuits for each input and output signal with large RF isolation. The coaxial connectors are SMA female. The top-side was fitted with a transparent plastic top inside the standard metal top so as to afford more protection under routine inspection. Leads were bonded from feedthrough terminals to the surface electrode busses via conductive silver paint.

Preliminary Experimental Results

Two devices were constructed. One was used to generate the 32-point sine and cosine chirp samples for premultiplication, and the other to convolve the multiplied input with the four 63-point chirp transversal filters.

Values used to construct the weights in the tapping structure are given in the first three columns of Table 1. The impulse response of a 32-point tapping structure is shown in Figure 6. A bulk-wave cancellation scheme was used to obtain this desired output, since some spurious signals were otherwise obtained with the electrode pattern on ST-cut quartz.

Values expected at the output terminals of the appropriately-summed convolver electrodes are given in the last three columns of Table 1 for the case when the input g_n is a constant = 1. This case simplifies the evaluation since two of the convolver operations and the entire post-multiplier operation become unnecessary. Values actually observed (after the loading interval) are shown in Figure 7, and the final summed output in Figure 8.

It is expected that the next generation device will have a better time-domain response such that the desired cancellation of the component responses will be more nearly perfect in the region following the dc peak.

Conclusions

The acoustic surface wave implementation of CZT convolvers and discrete chirp generators appears to hold much promise. Though the outputs observed to date are not yet satisfactory, they are expected to improve with refinement and practice. A similar surface wave generator/convolver utilizing nearly identical design techniques has been shown to perform with a very high level of accuracy [3].

Since the computation time of the algorithm is proportional to the number of samples, N, in the time signal, a time-domain implementation such as this one is expected to speed up calculation of the DFT by a factor of $\log_2 N$ relative to an FFT with identical capacity and speed.

The surface wave implementation is relatively small and light-weight, and has a low requirement for electrical power. The potentials for operating at higher frequencies and over longer acoustic paths (as compared to the design parameters reported here) point toward larger capacity, faster acoustic surface wave CZT devices in the future.

References

1. Rabiner, L. R., R. W. Schafer, and C. M. Rader, "The Chirp-Z Transform Algorithm," IEEE Trans. on Audio and Electroacoustics AU-17, No. 2, June 1969, pp. 86-92.

2. Whitehouse, H. J., J. M. Speiser, and R. W. Means, "High Speed Serial Access Linear Transform Implementations," Naval Undersea Center TN 1026 (also presented at All Applications Digital Computer (AADC) Symposium, Orlando, Florida), January 1973.

3. Alsup, J. M., "Acoustic Surface-Wave Golay-Coded Matched Filter," Naval Undersea Center TN 981, San Diego, CA, March 1973.

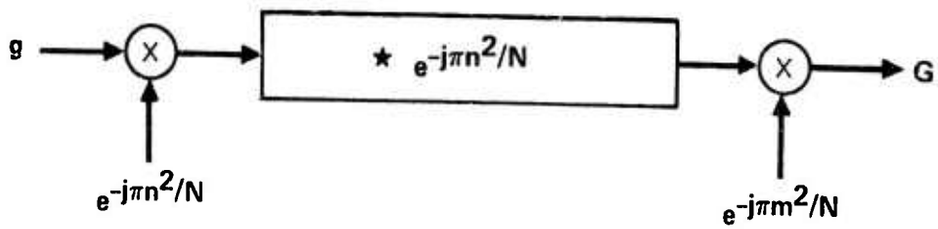


Fig. 1. Chirp Z Transform Implementation of the DFT

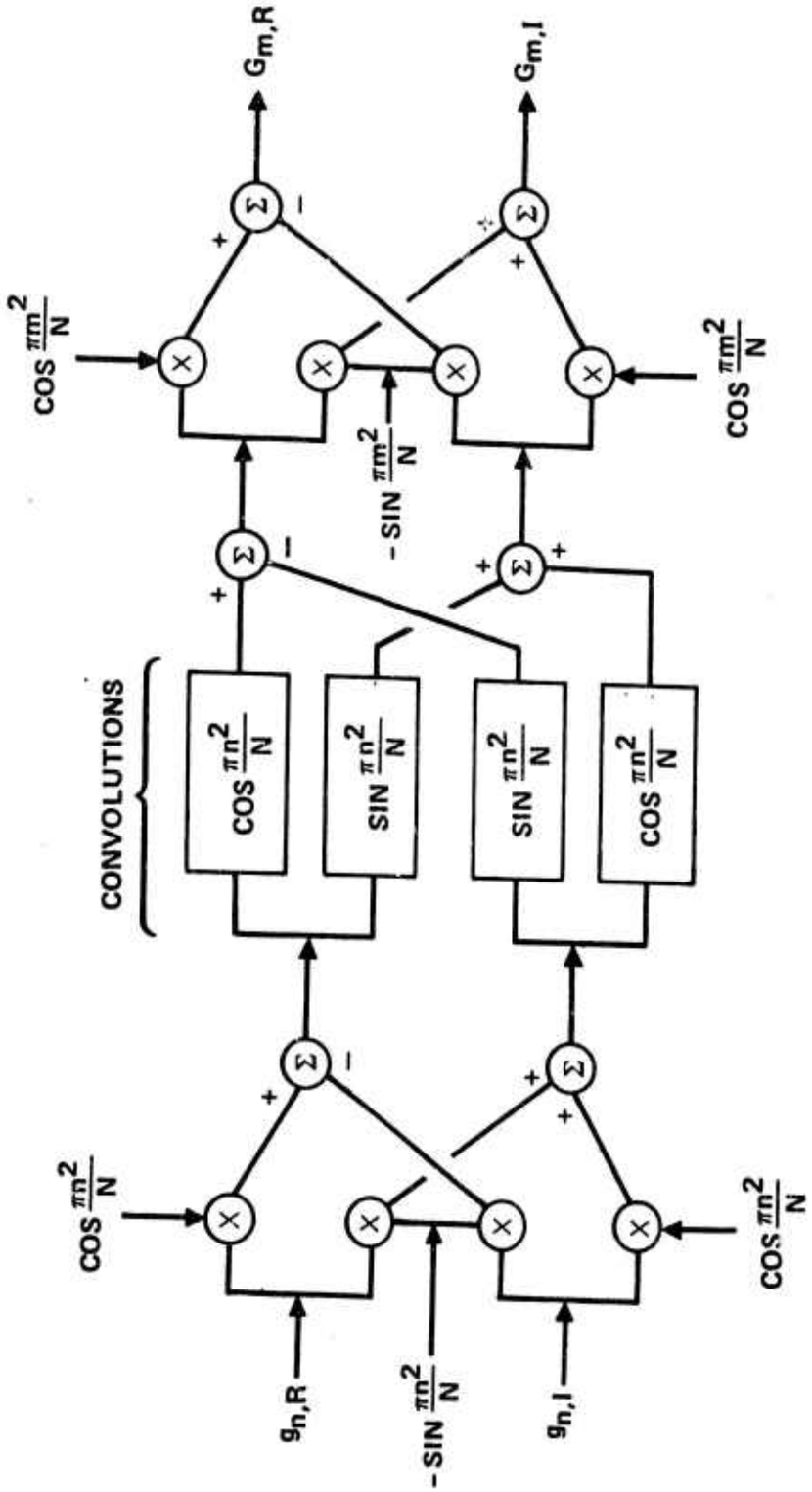
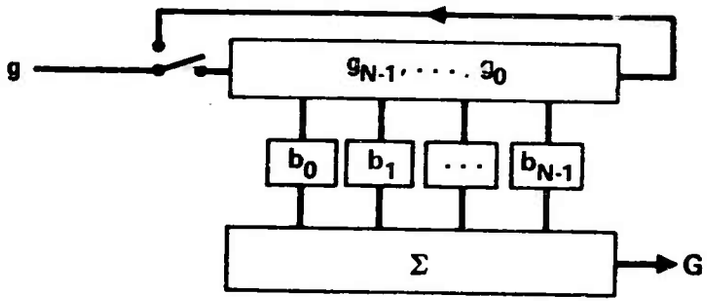


Fig. 3. DFT via CZT Algorithm with Parallel Implementation of Complex Arithmetic



$$\begin{pmatrix} G_0 \\ G_1 \\ \vdots \\ G_{N-1} \end{pmatrix} = \begin{pmatrix} b_{N-1} & b_{N-2} & \cdots & b_0 \\ b_0 & b_{N-1} & \cdots & b_1 \\ \vdots & & & \\ b_{N-2} & b_{N-3} & \cdots & b_{N-1} \end{pmatrix} \begin{pmatrix} g_0 \\ g_1 \\ \vdots \\ g_{N-1} \end{pmatrix}$$

Fig. 2. Circular Convolution

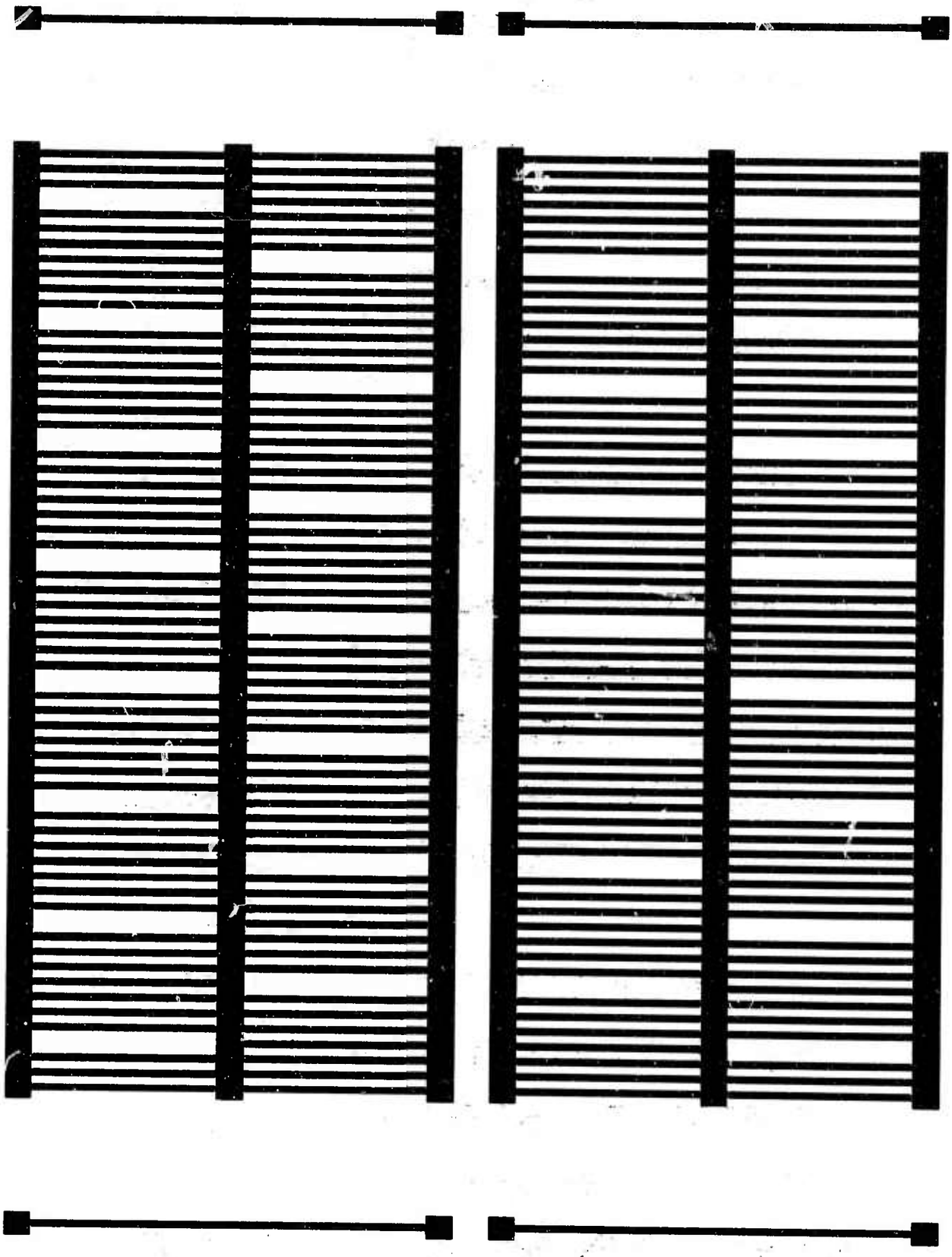


Fig. 4. CZT Transversal Filter

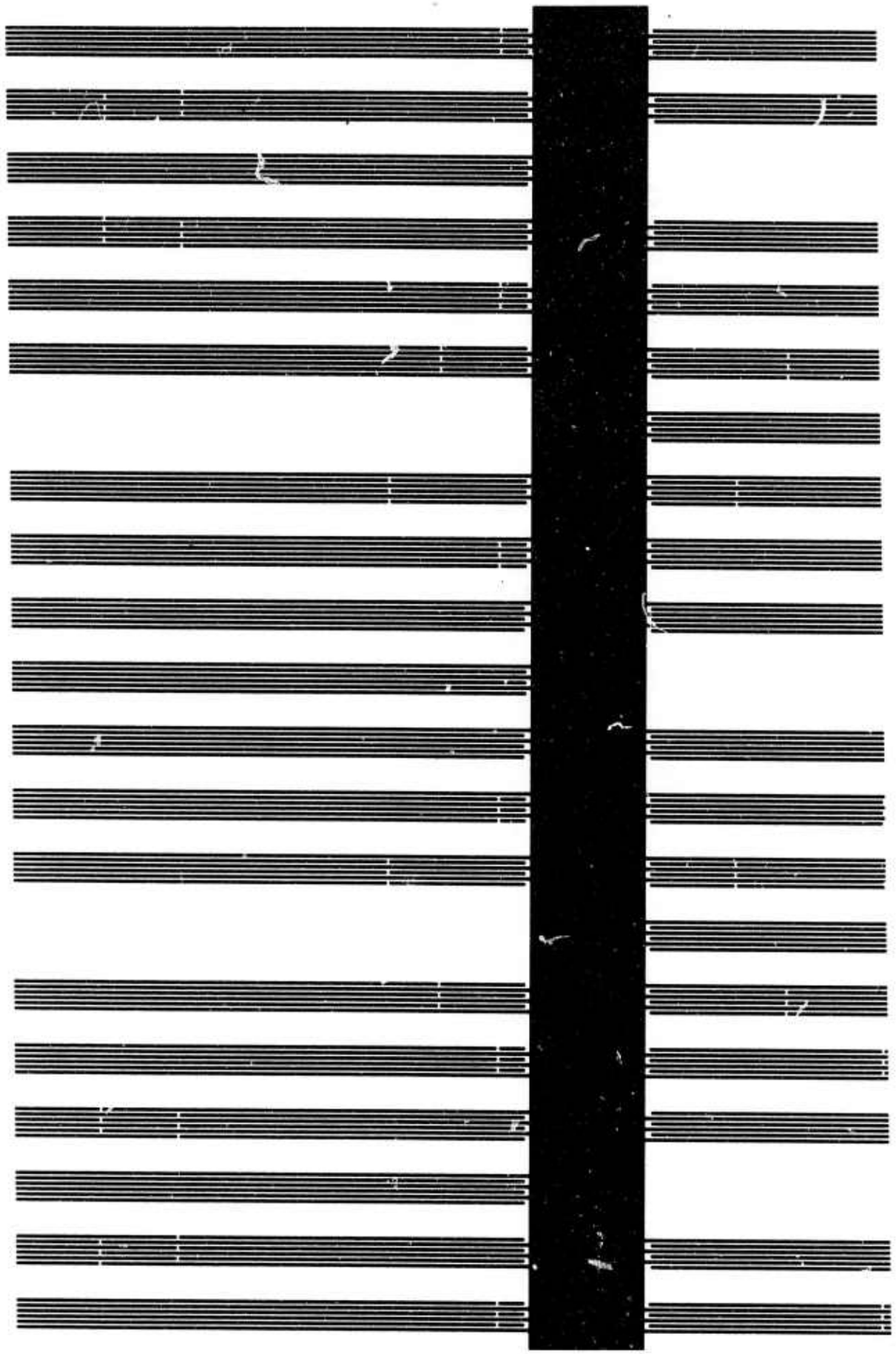


Fig. 5. CZT Transversal Filter, Close-Up

n	$\cos\pi n^2/32$	$\sin\pi n^2/32$	cos(32)*cos(63)	sin(32)*sin(63)	SUM
0	1.000	.000	18.8	13.2	32
1	.995	.098	0	0	0
2	.924	.383	2.2	-2.2	0
3	.634	.773	0	0	0
4	.000	1.000	0	0	0
5	-.773	.634	0	0	0
6	-.924	-.383	-3.3	3.3	0
7	.098	-.995	0	0	0
8	1.000	.000	-2.8	2.8	0
9	-.098	.995	0	0	0
10	-.924	-.383	3.3	-3.3	0
11	.773	-.634	0	0	0
12	.000	1.000	0	0	0
13	-.634	-.773	0	0	0
14	.924	.383	-2.2	2.2	0
15	-.995	-.098	0	0	0
16	1.000	.000	2.8	-2.8	0
17	-.995	-.098	0	0	0
18	.924	.383	-2.2	2.2	0
19	-.634	-.773	0	0	0
20	.000	1.000	0	0	0
21	.773	-.634	0	0	0
22	-.924	-.383	3.3	-3.3	0
23	-.098	.995	0	0	0
24	1.000	.000	-2.8	2.8	0
25	.098	-.995	0	0	0
26	-.924	-.383	-3.3	3.3	0
27	-.773	.634	0	0	0
28	.000	1.000	0	0	0
29	.634	.773	0	0	0
30	.924	.383	2.2	-2.2	0
31	.995	.098	0	0	0

Table 1. 32-Point CZT: Filter Tap Weights & Convolution Output Values for DC Input

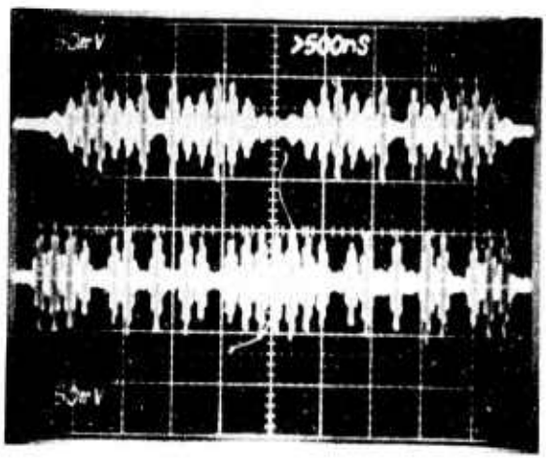


FIGURE 6. SURFACE WAVE CZT FILTER, 32-POINT IMPULSE RESPONSE: (a) $\sin \pi n^2/32$; (b) $\cos \pi n^2/32$.

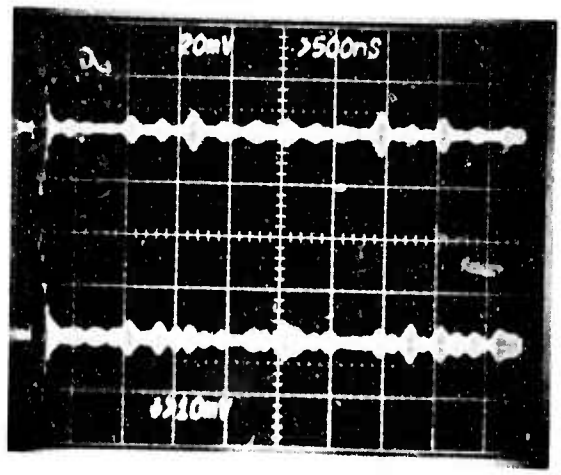


FIGURE 7. SURFACE WAVE CZT OUTPUT COMPONENTS FOR DC INPUT: (a) sin(32) * sin (63); (b) cos(32) * cos (63).

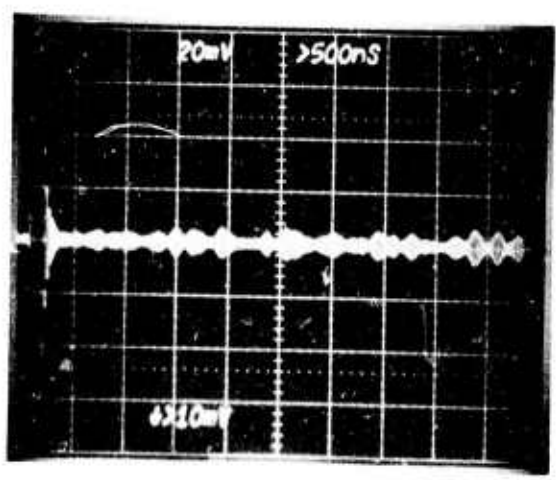


FIGURE 8. SURFACE WAVE CZT OUTPUT FOR DC INPUT.

APPENDIX D

REAL TIME DISCRETE FOURIER TRANSFORMS USING CTDS

REAL TIME DISCRETE FOURIER TRANSFORMS USING CHARGE TRANSFER DEVICES

R. W. Means
Naval Undersea Center

D. D. Buss
Texas Instruments

H. J. Whitehouse
Naval Undersea Center

ABSTRACT

A real time Discrete Fourier Transform (DFT) device is presented using bucket brigade devices. The chirp - z algorithm is used to compute the DFT. It required premultiplication of the time signal by a complex FM chirp, convolution with a complex FM chirp and postmultiplication by a complex FM chirp. Since the algorithm is complex it is implemented in real arithmetic by four convolvers operating in parallel. A two hundred stage bucket brigade transversal filter performs the convolution required. The signal is tapped with a source follower whose load determines the weighting coefficient. The computation time of the device is proportional to the number of samples, N, in the time signal. The computation time of the Fast Fourier Transform (FFT) is $N \log_2 N$ if N is a power of 2. The device is thus a factor of $\log_2 N$ faster than the FFT. It is also a low power, low weight device.

I. INTRODUCTION

Because of their analog nature, charge coupled devices (CCD's) offer the possibility of performing many of the signal processing functions which until now have fallen into the realm of digital filtering.[1] The discrete Fourier transform (DFT) of an electrical signal is one such linear filtering operation which readily lends itself to implementation with CCD's and is discussed in this paper.

The technique of using CCD's to perform the discrete Fourier transform (DFT) of an electrical signal was first proposed by Whitehouse, Speiser, and Means.[2] The technique they have proposed is called the chirp-Z transform (CZT) because the mathematical operations involved multiplication and convolution with chirp signals.[3]

The CZT was first conceived as an algorithm for the DFT on a digital computer. It never gained wide acceptance, however, because the number of digital operations required for its implementation is the same as for the Cooley-Tukey fast Fourier transform (FFT) algorithm. It can be viewed as three distinct operations, all of which can be performed with analog circuitry. (1) premultiplication by a chirp waveform, (2) convolution with a chirp waveform, and (3) post-multiplication by a chirp waveform. These three operations are illustrated in the block diagram of Figure 1.

The DFT of a band limited electrical signal g(t) is found by sampling the signal at N uniformly spaced instants of time to obtain the N points g_n. The DFT is then given by the definition

$$G_k = \sum_{n=o}^{N-1} g_n \exp\left[-\frac{i2\pi kn}{N}\right] \qquad o \leq k \leq N-1 \qquad (1)$$

The CZT algorithm for evaluation equation (1) is derived by making the substitution

$$2kn = k^2 + n^2 - (n-k)^2 \qquad (2)$$

into equation (1) to get

$$G_k = e^{-i\pi k^2/N} \sum_{n=0}^{N-1} e^{i\pi(n-k)^2/N} e^{-i\pi n^2/N} g_n \qquad (3)$$

Equation (3) suggests clearly the hardware implementation illustrated in Figure 1. This algorithm offers unique advantages when implemented using CCD transversal filters because all the processing can be performed in an analog format.

Section II of this paper gives preliminary experimental verification of the CZT concept. The demonstration is only approximate but it illustrates the simplicity of the approach. Section III describes the exact CZT implementations. Section IV concludes with the discussion of applications.

II. PRELIMINARY EXPERIMENTAL RESULTS

To demonstrate the concept of the CZT, the simplified system shown at the top of Figure 2 was implemented for approximating the power density spectrum. The filters used in this demonstration have impulse responses which approximate the functions.

$$h(t) = \cos \mu t^2 \qquad -\frac{T_d}{2} < t < \frac{T_d}{2} \qquad (4)$$

for the COS filter and

$$g(t) = \sin \mu t^2 \qquad -\frac{T_d}{2} < t < \frac{T_d}{2} \qquad (5)$$

for the SIN filter. They are bucket brigade device (BBD) filters which were developed for a spread-spectrum receiver.[4,5,6,7] The impulse responses and correlation responses of these filters are given in Figure 3.

In the demonstration, the chirp signal shown in Figure 2 chirps from 97.5 kHz to 102.5 kHz in 5 msec. and gives the approximate power density spectrum across this frequency band. The operation of the system can be understood in the following way. If the input signal is a single frequency sinusoid of frequency f_s, then the output of the mixer has a difference component which chirps from f_s -97.5 kHz to f_s -102.5 kHz and a sum component which is filtered out in a low-pass filter after mixing (not shown in Figure 2). The filters are clocked at 20 kHz and have

impulse responses which chirp from -5 kHz to +5 kHz in 10 msec.

The convolution operation is indicated schematically at the botton of Figure 2. The frequency of the filter impulse response is indicated by the line extending from -5 kHz to +5 kHz. The input signals are reversed in time as required by the convolution operation, and are shown for four values of f_s equally spaced across the chirp band: 97.5 kHz, 99.17 kHz, 100.83 kHz and 102.5 kHz. A correlation peak in the filter output occurs when the frequency of the time reversed input signal "lines up" with the frequency of the impulse response. At the time instant shown, the 97.5 kHz signal lines up with the filter and gives a correlation peak. Higher frequency signals result in correlation peaks which occur later in time by the amount of Δt given by

$$\Delta t = \frac{\mu}{\pi}(f_s - 97.5\text{kHz}) \qquad (6)$$

where $\frac{\mu}{\pi}$ is the slope of the frequency vs time curve as can be seen from eqs. (4) and (5).

Two filters are used to perform in-phase and quadrature processing, and the output signal which results from squaring and summing the filter outputs closely approximates the power density spectrum of the input signal across the 97.5 kHz to 102.5 kHz band.

The system of Figure 2 does not give the exact power density spectrum. There are small spurious sidelobes due to an approximation made that an up chirp is orthoganal to a down chirp. The system output also has a dependence upon the phase of the input signal. This dependence can be reduced by using filters of larger time-bandwidth product or by implementing the exact in-phase and quadrature processing to be discussed in Section III.

The actual operation of the system is illustrated in the series of photographs shown in Figure 4. The input is a single frequency sinusoid whose frequency takes on the values indicated in Figure 2, and the output shows a distinct correlation peak whose position scales uniformly with input frequency. The time axis calibration is 1.67 msec/cm, and the correlation peak shifts by 1 cm each time the input frequency increases by 1.67 kHz ($\mu/\pi = 10^6$ sec^{-2}). The

correlation peaks should all be of equal amplitude. However, because of the above mentioned dependence on input phase, there is a small random variation from one photograph to the next.

III. FOURIER TRANSFORM ANALYSIS

The exact DFT can be implemented by the hardware shown in Figure 5. The complex chirp Z algorithm as defined in equation (3) is implemented by real hardware operating in parallel. The transversal filters discussed in Section II can implement a cosine transform[8] which is a subset of the DFT. In order to illustrate this it is first necessary to discuss the symmetries involved in Fourier transforms.

If the input time signal g(t) is real, as it always is in signal processing applications, then the output Fourier transforms have the property that

$$G_k = \underline{G}^*_k \qquad (7)$$

This is an awkward symmetry to make use of in reducing data handling requirements. However, if the input data is real and symmetric, then the output Fourier transform is also real and symmetric.

This symmetric transform is sometimes referred to in the literature as the cosine transform. It is not uniquely defined and different authors have defined slightly different cosine transforms. There are two implicit symmetrizations possible on the input data: (1) The data set of length N is reflected about a data point. This will result in a data set of length 2N-1; (2) The input data set of length N is reflected so that the resultant data set is of length 2N. An example should clarify these two casses.

1) Original data set: [1,2,3,3,4]
   Symmetrized data set: [4,3,3,2,1,2,3,3,4]

2) Original data set: [1,2,3,3,4]
   Symmetrized data set: [4,3,3,2,1,1,2,3,3,4]

The Naval Undersea Center is interested in implementing the cosine transform because it is involved in a program of television band width reduction. The ordinary discrete Fourier transform has several faults. Since it is really a Fourier transform of a periodically repeated signal, it may have artificial discontinuities at the boundaries. If the signal is made symmetric then the discontinuities dissappear and the Fourier transform will thus converge more rapidly. It will be shown later that making the signal symmetric does not take any hardware. In fact it reduces the amount of hardware required to perform a Fourier Transform.

In order to demonstrate this it is advantageous to go through several stages of analysis. Consider a continuous Fourier transform of a finite time signal g(t)

$$G(\omega) = \int_0^T g(t)e^{-i\omega t}\,dt \tag{8}$$

Define the symmetrized time signal as

$$\hat{g}(t) \equiv \begin{matrix} g(t) & 0 \le t \le T \\ g(-t) & -T \le t \le 0 \end{matrix} \tag{9}$$

Then the Fourier transform of $\hat{g}(t)$ is

$$\hat{G}(\omega) = \int_{-T}^{T} \hat{g}(t)\cos\omega t\,dt \tag{10}$$

Since $\hat{g}(t)$ is real and symmetric then $G(\omega)$ is real and symmetric and thus

$$\begin{aligned} \hat{G}(\omega) &= \int_{-T}^{T} \hat{g}(t)\cos\omega t\,dt \\ &= 2\int_0^T g(t)\cos\omega t\,dt \end{aligned} \tag{11}$$

For future comparison it is convenient to write this as

$$\hat{G}(\omega) = 2\,\mathrm{Re}\int_0^T g(t)e^{-i\omega t}\,dt \tag{12}$$

Notice that the Fourier transform of the symmetric time signal $\hat{g}(t)$ has been derived by only working with the non-symmetric time function g(t) and that only an integral of length T was required.

Let us consider now a sampled signal g_n of N points and its discrete Fourier transform. Define the symmetric function

$$\hat{g}_n \equiv \begin{cases} g_{n-1} & n \geq 1 \\ g_{-n} & n \leq 0 \end{cases} \qquad -N+1 \leq n \leq N \tag{13}$$

This function can be looked on as symmetric about n = 1/2. Its "Fourier" transform is thus

$$\hat{G}_k = \sum_{n=-N+1}^{N} \hat{g}_n \, e^{\frac{-2\pi i k(n-1/2)}{2N}} \qquad -N+1 \leq k \leq N \tag{14}$$

Since $\hat{g}_n$ is symmetric, then Im $\hat{G}_k = 0$ and $\hat{G}_k = \hat{G}_{k+1}$ and it is only necessary to compute half the set.

$$\hat{G}_k = \sum_{n=-N+1}^{N} \hat{g}_n \, e^{\frac{-2\pi i k(n-1/2)}{2N}} \qquad o \leq k \leq N-1 \tag{15}$$

By rearranging the summing order and using the fact that $\hat{g}_n = g_{n-1}$ for $n \geq 1$, this can be shown to be

$$\hat{G}_k = 2 \sum_{n=o}^{N-1} g_n \cos\left[\frac{2\pi k(n+1/2)}{2N}\right] \qquad o \leq k \leq N-1 \tag{16}$$

This can be written in the form

$$\hat{G}_k = 2 \, \mathrm{Re} \sum_{n=o}^{N-1} g_n \, e^{\frac{2\pi i k(n+1/2)}{2N}} \qquad o \leq k \leq N-1 \tag{17}$$

Notice that only the original sampled time function $\{g_n\}$ is used and that the sum is only over N points. Equation 17 can be implemented by a chirp Z transform by substituting the identity.

$$2k(n+1/2) = k^2 + (n+1/2)^2 - (k-(n+1/2))^2 \tag{18}$$

This gives

$$\hat{G}_k = 2 \, \mathrm{Re} \, e^{\frac{-i\pi k^2}{2N}} \sum_{n=o}^{N-1} e^{+i\pi(k-(n+1/2))^2/2N} * \\ * e^{-i\pi(n+1/2)^2/2N} g_n \tag{19}$$

This cosine transform can be implemented by the bucket brigade devices by letting N = 199 except for a factor of 2 in the total length of the filter. This implies that only the first 100 cosine transform coefficients will be computed and that the data is defined to be equal to zero after n = 100.

IV. APPLICATIONS

The CZT algorithm is conceptually a real time algorithm for a sampled data sytem. The set of points $\{g_n\}$ can be considered as a time sampled function of a continuous function g(t) of length T. The transversal filter of the chirp Z transform will have a basic clock frequency equal to the sampling frequency. It has 2N (or 2N-1) stages so it is approximately twice as long as the input signal. A time T must pass before the set g_n fills the first half of the transversal filter. Thereafter, one DFT coefficient is computed for every time step. This is what is meant by a real time device, i.e. for a data sequence of length N it takes N clock periods to compute the DFT.

By this definition the chirp Z algorithm can be compared with other algorithms. The basic algorithm as defined in equation (1) takes N^2 steps to compute. If N is a power of 2, then the Cooley-Tukey algorithm called the Fast Fourier Transform (FFT) required $N \log_2 N$ steps to compute. The transversal filter implementation of the chirp Z algorithm achieves its high speed not by reducing the number of multiples like the FFT but by putting the equivalent of N parallel multipliers all on the same chip. This reduces the number of computational steps from N^2 to N.

A real time Fourier transform device will find many applications in radar, sonar and pattern recognition. One of the most promising is the bandwidth reduction of television.[9]

ACKNOWLEDGEMENTS

The construction of the BBD filters discussed in Section II was sponsored by Rome Air Development Center under contract # F30602-73-C-0027 monitored by Charles N. Meyer of RADC. The implementation of the CZT demonstration was partially sponsored by the Army Electronics Command under contract # DAAB07-73-C-0351 monitored

by Robert Sproat of Ft. Monmouth and partially sponsored by the Defense Advanced Research Projects Agency under order number 2303, program code number 3G10 monitored by Colonel H. M. Federhen.

BIBLIOGRAPHY

1. B. Gold and C. M. Rader, Digital Processing of Signals, McGraw-Hill, 1966.
2. H. J. Whitehouse, J. M. Speiser and R. W. Means, "High Speed Serial Access Linear Transform Implementations," presented at the AII Applications Digital Computer Symposium, Orlando, Florida, 23-25 January 1973 NUC TN 1026.
3. L. R. Rabiner, R. W. Shafer, and C. M. Rader, "The Chirp Z Transform Algorithm," IEEE Trans. Audio and Electroacoustics AU 17, 86 (1969).
4. For a discussion of BBD technology see "Practical Considerations for Analog Operation of Bucket-Brigade Circuits," W. J. Butler, M. B. Barron, and C. M. Puckette, IEEE Jol. Solid-State Circuits SC-8, April 1973, pp. 157-168.
5. For a discussion of BBD transversal filters see "Transversal Filtering Using Charge-Transfer Devices," D. D. Buss, D. R. Collins, W. H. Bailey, and C. R. Reeves, IEEE Jol. Solid-State Circuits, SC-8, April 1973, pp.138-146.
6. Contract # F30602-73-C-0027, Final Report, Part II.
7. "Applications of Charge Transfer Devices to Communication," D. D. Buss, and W. H. Bailey. This Conference
8. N. Ahmed, T. Natarajan, and K. R. Rao, "On Image Processing and a Discrete Cosine Transform," to be published.
9. R. W. Means and H. J. Whitehouse, "Image Transmission Via Spread Spectrum Techniques," NUC Technical Report for ARPA Order Number 2303, July 1973.

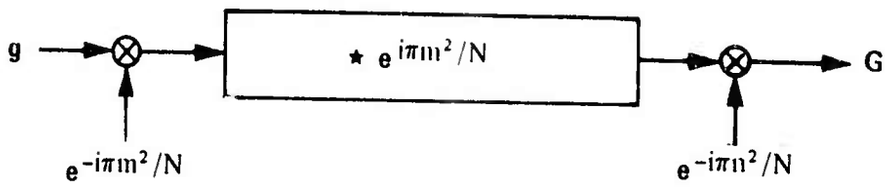
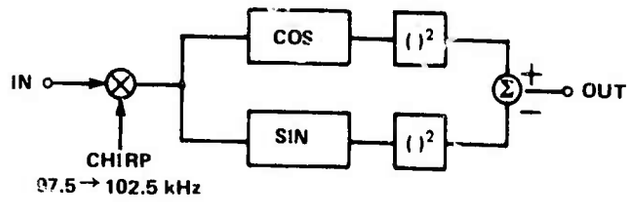


Fig. 1. Chirp Z-transform Implementation of the DFT



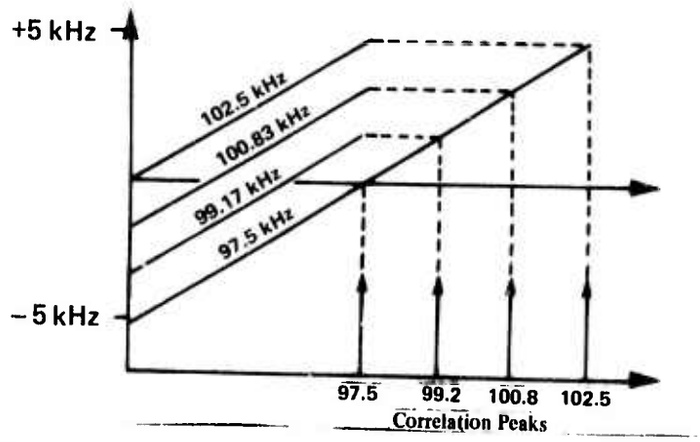


Fig. 2. TOP: Block D agram of a Configuration for Approximating the Power Density Sprectrum. The filters marked COS and SIN are clocked at 20 kHz so that their impulse responses chirp from -5 kHz to +5 kHz. A low pass filter which is not shown follows the mixing operation and eliminates the sum frequency components.
BOTTOM: Schematic of the Convolution Operation Performed Within the Filters. (See text).

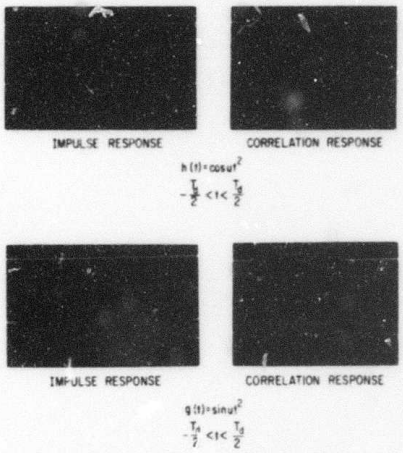


Fig. 3 The impulse response and correlation response of the two filters required for the system of Fig. 2. These filters are each 200 stages long and are both integrated in a single BBD IC.

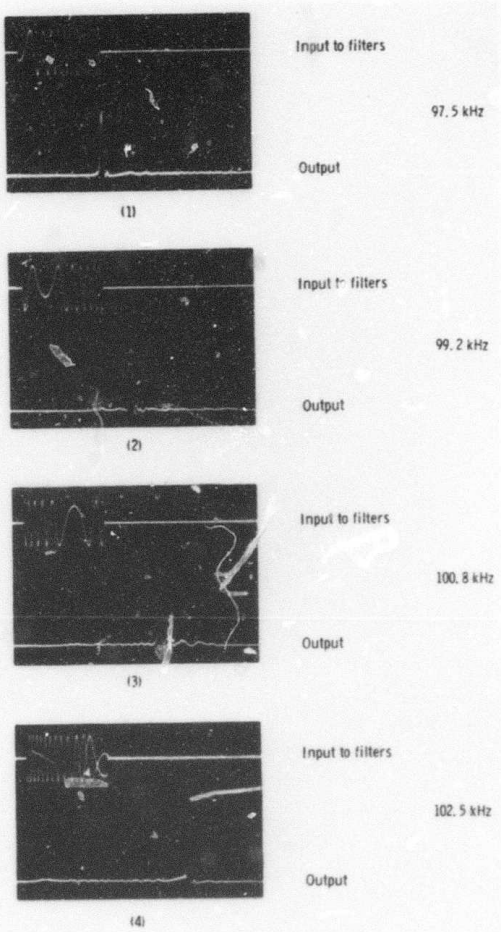


Fig. 4. Power density spectrum output for sinusoidal input

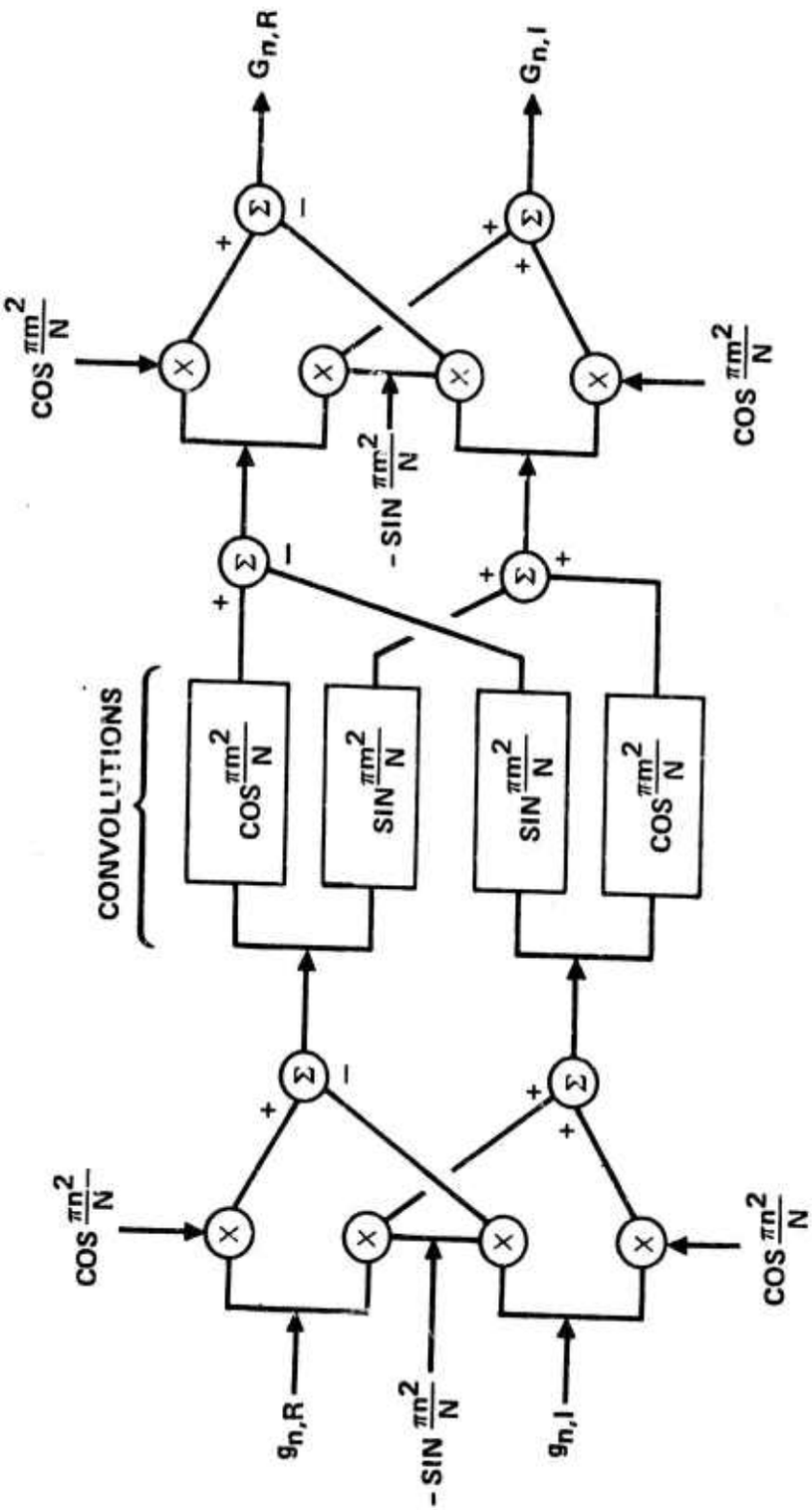


Fig. 5 DFT via CZT Algorithm with Parallel Implementation of Complex Arithmetic

APPENDIX E

APPLICATION OF IMAGE EXTRAPOLATION

Transform Spectrum Extrapolation

by

William K. Pratt
Electrical Engineering Department
Image Processing Institute
University of Southern California
Los Angeles, California 90007

October 1973

Submitted to

IEEE Transactions on Information Theory

This work was supported by the Advanced Research Projects Agency of the Department of Defense and monitored by the Air Force Eastern Test Range under Contract F08606-72-C-0008 and by the Naval Undersea Center, San Diego, under Contract N00123-73-C-1507.

Abstract

A generalized technique is developed for estimating the missing spectral components of the mathematical transformation of a data vector. The method has application in transform data coding systems and time series analysis problems.

The spectrum extrapolation matrix operator is derived upon principles of minimum mean square error estimation. Performance of the extrapolation method is analyzed, and experimental results of the application of the method to one and two dimensional signals is presented.

1. Introduction

In transform data coding systems one of the coding techniques is to transmit only those data transform coefficients of low frequency or those components of low magnitude [1]. At the decoder it has been common practice to insert zeros for the missing components prior to the inverse transformation operation that reconstructs the data. However, it is known that for all transformations, other than the Karhunen-Loeve transform, the coefficients are partially correlated. Hence, it should be possible to utilize knowledge of this correlation to estimate the missing coefficients rather than arbitrarily setting their values to zero. Another related problem occurs in time series analysis in which it is often required to estimate the power spectral density of a process given only a few points of the autocorrelation function [2]. A general solution to this class of problems has been formulated within the framework of linear minimum mean square estimation theory.

2. Generalized Spectra

Consider an N element vector $\underline{f}$ which is a sample of a random process with zero mean and known covariance matrix $\underline{K}_f$. This vector undergoes a linear transformation defined by the N x N element operator matrix $\underline{A}$. Typical transformations of interest include the Fourier, Cosine, Hadamard, Haar, Slant, and Karhunen-Loeve transforms [3-8]. The N element vector $\underline{f}$ denotes the one dimensional transformation of $\underline{f}$ as given by

$$\underline{f} = \underline{A}\,\underline{f} \tag{1}$$

where the elements of $\underline{f}$ represent the spectral components of an expansion of $\underline{f}$ in terms of a set of basis vectors which are the rows of $\underline{A}$. Since $\underline{f}$ is zero mean, $\underline{f}$ also possess a zero mean. Also, for a unitary transformation it is easily seen that the covariance matrix of the transformed vector is

$$\underline{K}_f = \underline{A}\,\underline{K}_f\,\underline{A}^{-1} \tag{2}$$

3. Spectrum Extrapolation-Noise Free Condition

Figure 1a contains a diagram illustrating the generation of the generalized spectral components of a signal, the selection or truncation of these components, and the estimation of the discarded components. In the diagram the complete vector of spectral components $\underline{f}$ is operated upon by an M x N (M< N) selection matrix $\underline{S}$ that extracts certain elements of $\underline{f}$ and records them in an M element vector $\underline{f}_T$. The selection matrix contains a unit element at S(i, j) to transfer the jth element of $\underline{f}$ to the i th element of $\underline{f}_T$. This selection matrix represents the sample selector of a transform data coding system, or it can be considered to represent the incomplete knowledge of a correlation function in the problem of power spectral density estimation.

In the diagram of figure 1a, the vector $\underline{f}_T$ is multiplied by an N x M estimation matrix $\underline{W}$ which provides an estimate, $\hat{\underline{f}}$, of the complete set of spectral components of $\underline{f}$. An inverse unitary transformation $\underline{A}^{-1}$ then reconstructs an estimate, $\hat{\underline{f}}$, of the signal vector, $\underline{f}$. In essence, the estimator linearly extrapolates the values of the observation of known spectral components to determine the unknown spectral values. The extrapolation is meaningful because of the information provided by the expected correlation of the elements of the observation.

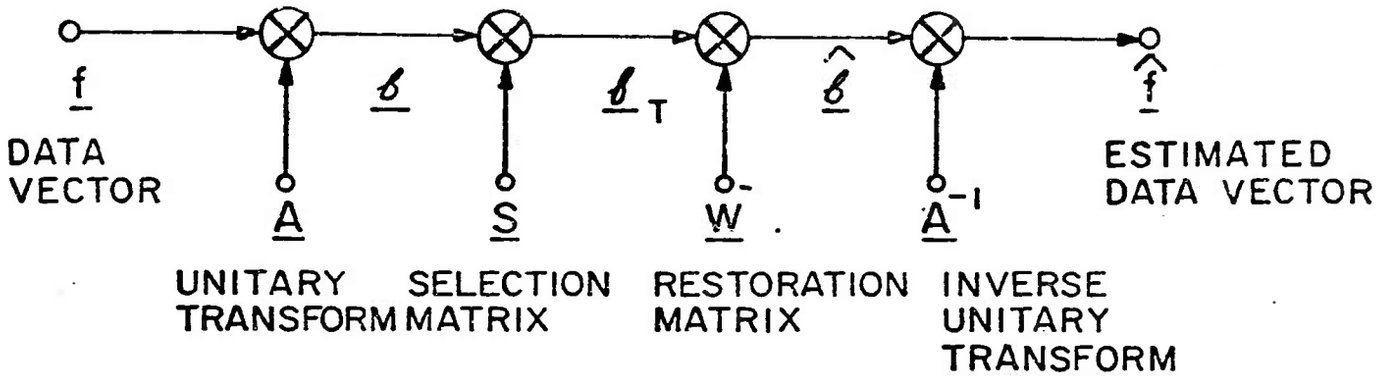
The estimator matrix $\underline{W}$ is chosen to minimize the mean square error, ε, between the signal vector and its estimate as defined by*

$$\varepsilon = \mathrm{tr}\left[E\{(f-\hat{f})(f-\hat{f})^{*T}\}\right] \qquad (3)$$

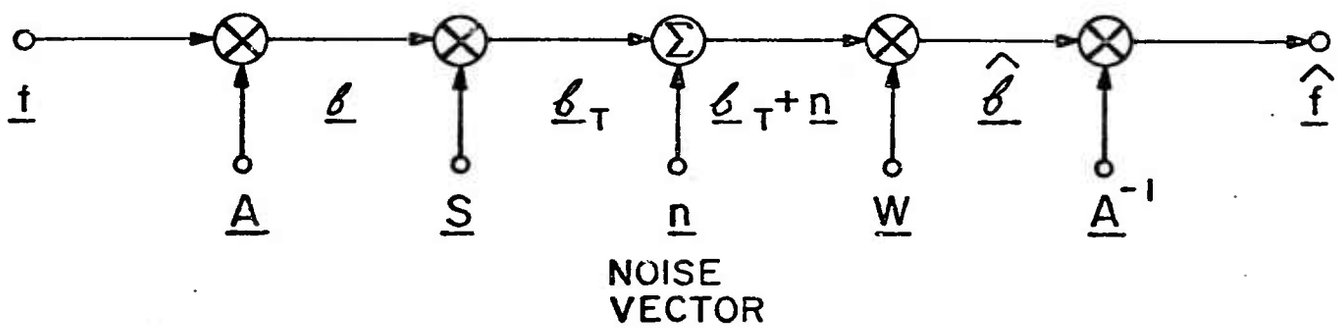
The optimum choice of $\underline{W}$ to minimize the mean square error can be found by forcing the dynamic error $(\underline{f}-\hat{\underline{f}})$ to be orthogonal to the observation $\underline{f}_T$ [9].

---

notation: E{ . } = expected value; tr=trace of a matrix; * = complex conjugate; T = transpose.



(a) Noise-free estimation



(b) Additive noise estimation

Figure 1. Transform spectrum extrapolation operations.

Thus, by setting

$$E\left\{ (\underline{f}-\underline{\bar{f}})(\underline{f}_T)^{*T} \right\} = \underline{0} \tag{4}$$

where $\underline{0}$ is an NxM matrix of zeros, one obtains

$$E\left\{ (\underline{f}-\underline{A}^{-1}\,\underline{W}\,\underline{S}\,\underline{A}\,\underline{f})\,(\underline{S}\,\underline{A}\,\underline{f})^{*T} \right\} = \underline{0} \tag{5}$$

This leads directly to the optimum solution

$$\underline{W} = \underline{A}\,\underline{K}_f\,\underline{A}^{-1}\,\underline{S}^T\,[\underline{S}\,\underline{A}\,\underline{K}_f\,\underline{A}^{-1}\,\underline{S}^T]^{-1} \tag{6}$$

Premultiplying $\underline{W}$ by the selection matrix $\underline{S}$ yields

$$\underline{S}\,\underline{W} = \underline{I}_M \tag{7}$$

where $\underline{I}_M$ denotes an MxM identity matrix. Since

$$\hat{\underline{f}} = \underline{W}\,\underline{f}_T \tag{8}$$

it is clear from equation (6) that

$$\underline{S}\,\hat{\underline{f}} = \underline{S}\,\underline{W}\,\underline{f}_T = \underline{f}_T \tag{9}$$

Thus, the estimator simply copies the elements of $\underline{f}_T$ into the appropriate elements of $\hat{\underline{f}}$. The remaining elements of $\hat{\underline{f}}$ are obtained by a linear combination of the elements of $\underline{f}_T$. Examination of the extrapolation matrix computed from equation (6) indicates that the first M rows of $\underline{W}$ form an identity matrix. The remaining N-M rows of $\underline{W}$ contain elements whose weightings are proportional to the correlation between the coefficient to be estimated and the spectral observations. For a Karhunen-Loeve transformation, all transform coefficients are uncorrelated, and hence spectrum extrapolation is not applicable. In this instance the bottom N-M rows of $\underline{W}$ are all composed of zero elements.

The spectral estimation problem diagrammed in figure 1a could have been formulated initially to estimate the signal $\underline{\hat{f}}$ directly from the spectral observation $\underline{f}_T$. Let this estimator be denoted as $\underline{U}$ where

$$\underline{\hat{f}} = \underline{U}\,\underline{f}_T \tag{10}$$

However, since $\underline{A}$ is a unitary matrix, the measure of the error is preserved by the transformation, and it is easy to show that

$$\underline{U} = \underline{A}^{-1}\,\underline{W} \tag{11}$$

Thus, the original solution is optimum.

4. Spectrum Extrapolation - Additive Noise

In physical situations the observation usually is contaminated by noise. If the noise is additive, as in figure 1b, the optimum estimator that minimizes the mean square error, ε, will, in effect, perform an additional task of noise filtering. Applying the orthogonality principle, one obtains

$$E\left\{ (\underline{f}-\underline{\hat{f}})(\underline{v})^{*T} \right\} = \underline{0} \tag{12a}$$

or

$$E\left\{ \left[\underline{f}-\underline{A}^{-1}\,\underline{W}\,(\underline{n}+\underline{S}\,\underline{A}\,\underline{f})\right]\left[\underline{n}^{*T} + \underline{f}^{*T}\,\underline{A}^{-1}\,\underline{S}^{T}\right]\right\} = \underline{0} \tag{12b}$$

For the usual case of zero mean noise in which the signal and noise are uncorrelated, the optimum estimator is found to be

$$\underline{W} = \underline{A}\,\underline{K}_f\,\underline{A}^{-1}\,\underline{S}^{T}\left[\underline{S}\,\underline{A}\,\underline{K}_f\,\underline{A}^{-1}\,\underline{S}^{T} + \underline{K}_n\right]^{-1} \tag{13}$$

where $\underline{K}_n$ is the noise covariance matrix.

5. Computational Considerations

In the noise free case of spectrum extrapolation, it is possible to reformulate the expression for the extrapolation matrix of equation (6) to permit computation using a pseudoinverse [10] algorithm. Let

$$\underline{B} \equiv \underline{S}\,\underline{A}\,\underline{H}_f \tag{14}$$

where $\underline{H}_f$ is obtained by a factorization of the covariance matrix $\underline{K}_f$. That is

$$\underline{K}_f = \underline{H}_f \underline{H}_f^T \tag{15}$$

A general factorization technique is given by

$$\underline{H}_f = \underline{E}\,\underline{\Lambda}_s\,\underline{M} \tag{16}$$

where $\underline{E}$ is a matrix containing the eigenvectors of $\underline{K}_f$ as column vectors, $\underline{\Lambda}_s$ in a diagonal matrix whose elements are the square roots of the eigenvalues of $\underline{K}_f$, and $\underline{M}$ is an arbitrary unitary matrix, which may be chosen to force $\underline{H}_f$ to be of lower triangular form. With this factorization the extrapolation matrix expression becomes

$$\underline{W} = \underline{A}\,\underline{H}_f\,\underline{B}^T\,(\underline{B}\underline{B}^T)^{-1} \tag{17}$$

where it is observed that the matrix $\underline{P} \equiv \underline{B}^T(\underline{B}\underline{B}^T)^{-1}$ can be computed as the pseudoinverse of $\underline{B}$. Algorithms exist for the efficient computation of a matrix pseudoinverse [11].

6. Two Dimensional Formulation

The spectrum extrapolation technique can be extended to two dimensional transforms quite easily. Consider a PxP data matrix $\underline{F}$ which is column

scanned to obtain the $N=P^2$ data vector $\underline{f}$. Substitution may then be made directly into equation (1), to apply the subsequent one dimensional processing algorithm. A special situation of interest occurs when the data covariance matrix, transform matrix, and selection matrix are all of separable form such that the following direct product relations hold

$$\underline{K}_f = \underline{K}_{fx} \times \underline{K}_{fy} \tag{18a}$$

$$\underline{A} = \underline{A}_x \times \underline{A}_y \tag{18b}$$

$$\underline{S} = \underline{S}_x \times \underline{S}_y \tag{18c}$$

where $\underline{K}_{fx}$ and $\underline{K}_{fy}$ are (PxP) row and column covariance matrices, $\underline{A}_x$ and $\underline{A}_y$ are row and column transform matrices, and $\underline{S}_x$ and $\underline{S}_y$ are row and column selection matrices. For the noise-free case it is easily shown that

$$\underline{W} = \underline{W}_x \times \underline{W}_y \tag{19}$$

where

$$\underline{W}_x = \underline{A}_x \underline{K}_{fx} \underline{A}_x^{-1} \underline{S}_x^T [\underline{S}_x \underline{A}_x \underline{K}_{fx} \underline{A}_x^{-1} \underline{S}_x^T]^{-1}$$

$$\underline{W}_y = \underline{A}_y \underline{K}_{fy} \underline{A}_y^{-1} \underline{S}_y^T [\underline{S}_y \underline{A}_y \underline{K}_{fy} \underline{A}_y^{-1} \underline{S}_y^T]^{-1}$$

In this case spectrum extrapolation can be sequentially performed on rows and columns of the truncated two dimensional array of transform coefficients, thus permitting a significant reduction in computational complexity. Unfortunately, the two dimensional extrapolation matrix of equation (13) for a noisy observation is not separable when the separability conditions of equation (18) hold, even if the noise is separable and uncorrelated.

7. Performance Evaluation

The usual alternative to spectrum extrapolation is simply to set the values of the discarded spectral components to zero and perform an inverse transformation. For such a strategy the reconstruction matrix is simply $\underline{W} = \underline{S}^T$ and the estimated vector becomes

$$\hat{\underline{f}} = \underline{A}^{-1} \underline{S}^T \underline{S} \underline{A} \underline{f} \tag{20}$$

Substitution into equation (3) yields the sub-optimal mean square error

$$\delta_{so} = \text{tr} [\underline{K}_f (\underline{I}_N - \underline{A}^{-1} \underline{S}^T \underline{S} A)] \tag{21a}$$

or

$$\delta_{so} = \text{tr} [\underline{K}_f (\underline{I}_N - \underline{S}^T \underline{S})] \tag{21b}$$

If the optimal reconstruction matrix of equation (6) or equation (13) is employed, the resulting mean square error is found to be

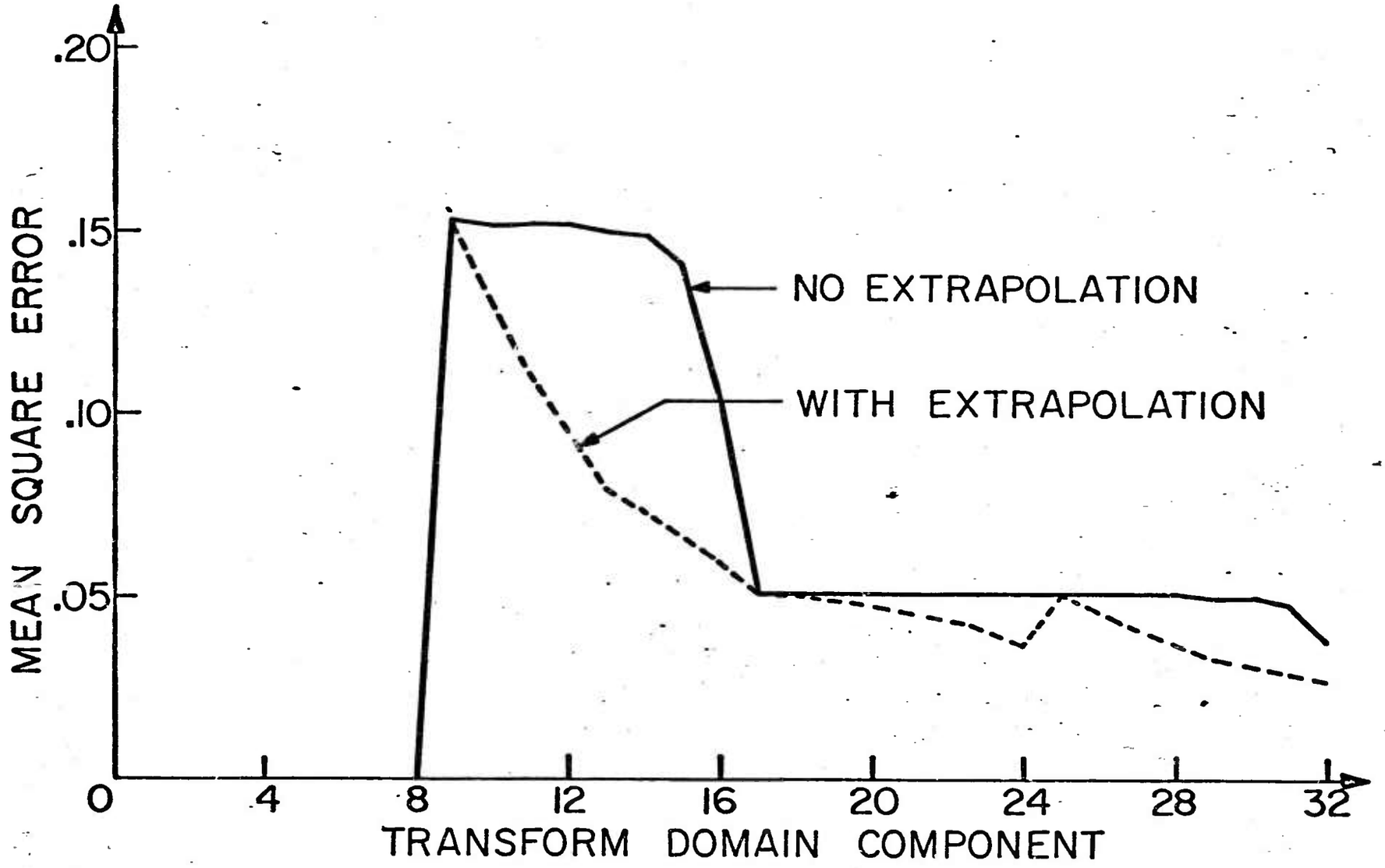
$$\delta_o = \text{tr} [\underline{K}_f (\underline{I}_N - \underline{W} \underline{S})] \tag{22}$$

Therefore, a reduction in mean square error by the amount

$$\delta_D = \delta_{so} - \delta_o = \text{tr} [\underline{K}_f (\underline{W} - \underline{S}^T) \underline{S}] \tag{23}$$

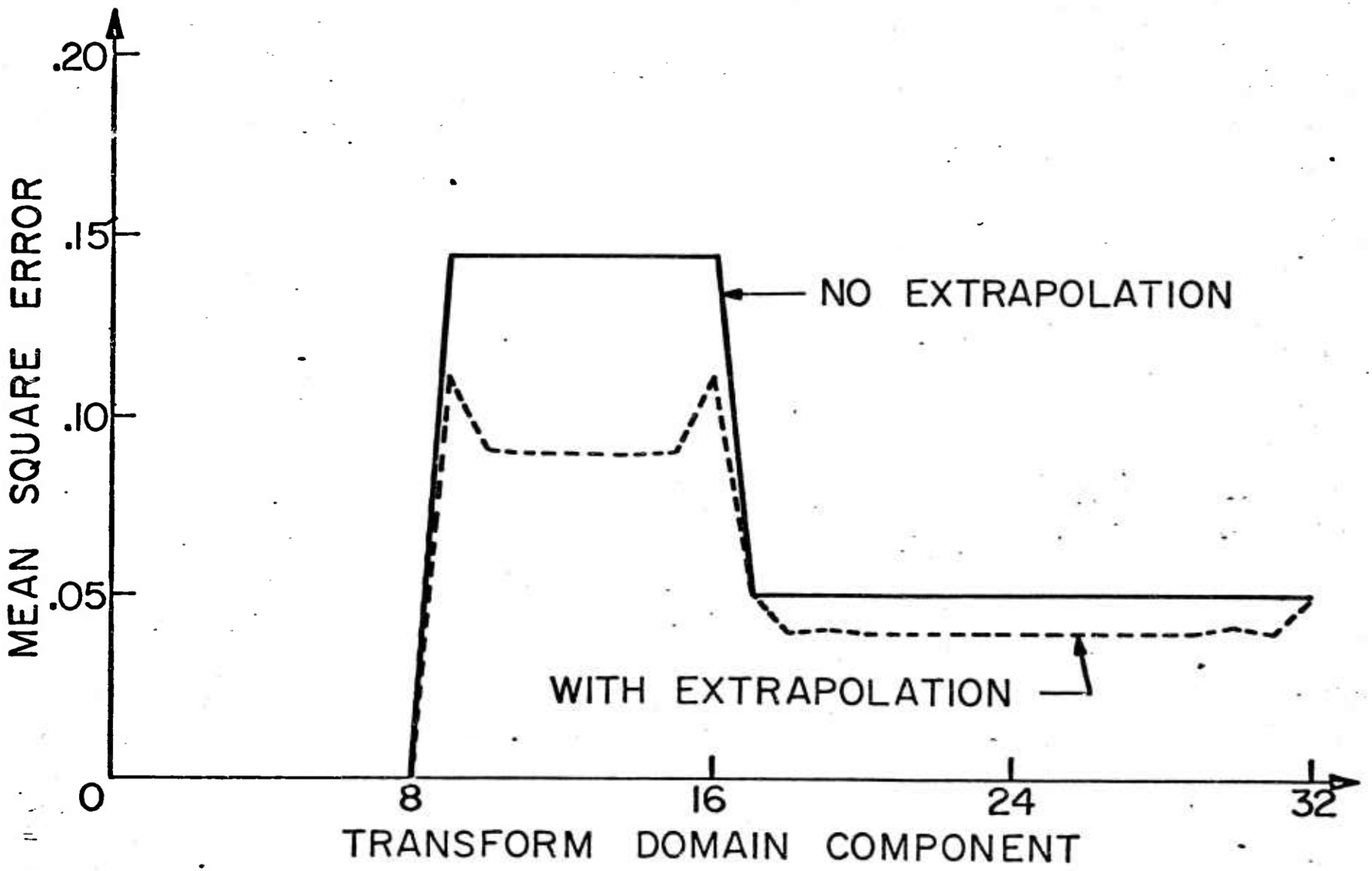
can be realized by spectrum extrapolation.

Figure 2 contains plots of the mean square error of transform components with and without spectrum extrapolation for several transforms. In this example the data vector is of length 32 and the last 24 elements have been truncated. The data is modelled as a first order Markov process with an adjacent element correlation of 0.9. The figures indicate significant reductions in mean square error are obtained for the Hadamard, Haar and Slant transforms. No error reduction is possible with the Karhunen-Loeve transform since the transform coefficients are uncorrelated, and there is no information available for extrapolation. There



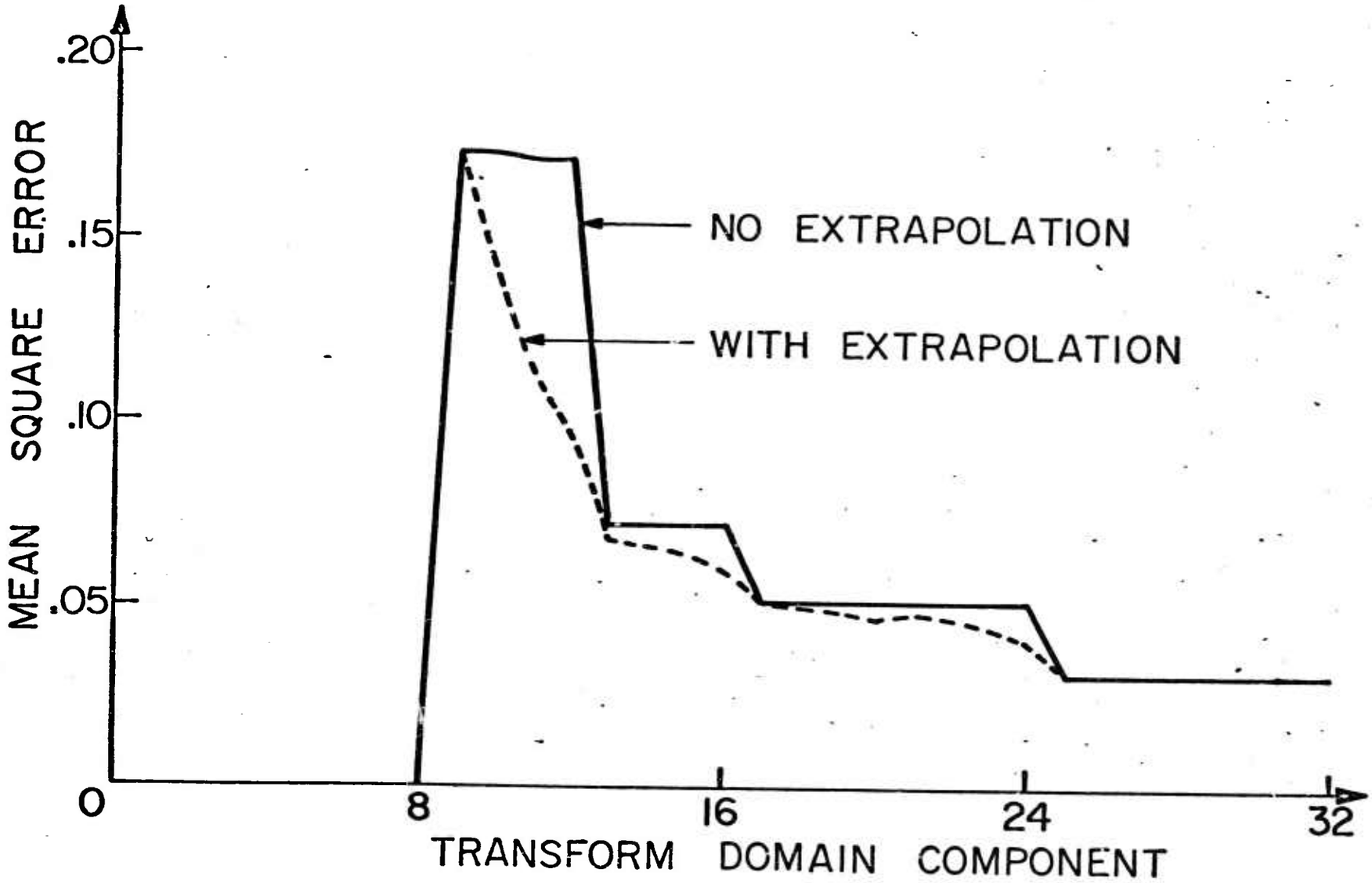
(a) Hadamard transform

Figure 2. Mean square reduction with spectrum extrapolation for N=32 and M=8 for a Markov process source with $\rho = 0.95$.



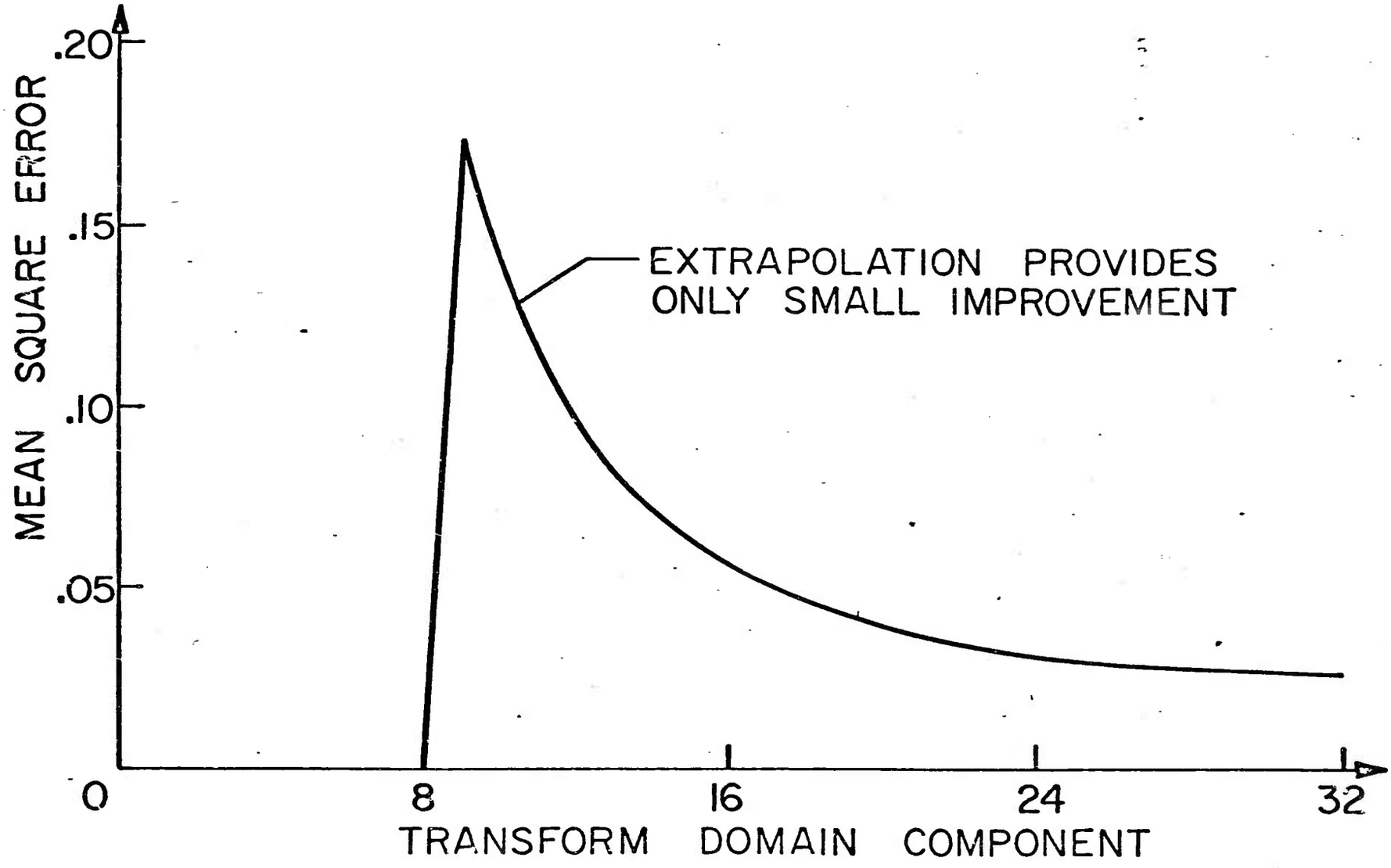
(b) Haar transform

Figure 2. (continued)



(c) Slant transform

Figure 2. (continued)



(d) Cosine transform

Figure 2. (continued)

(a) original

(b) 4:1 zonal selection

(c) 4:1 zonal selection spectrum extrapolation

(d) 10:1 zonal selection

(e) 10:1 zonal selection spectrum extrapolation

Figure 4: Hadamard transform spectrum extrapolation examples.

is almost no improvement obtained with the Cosine transform because of its close similarity to the Karhunen-Loeve transform for Markov process data.

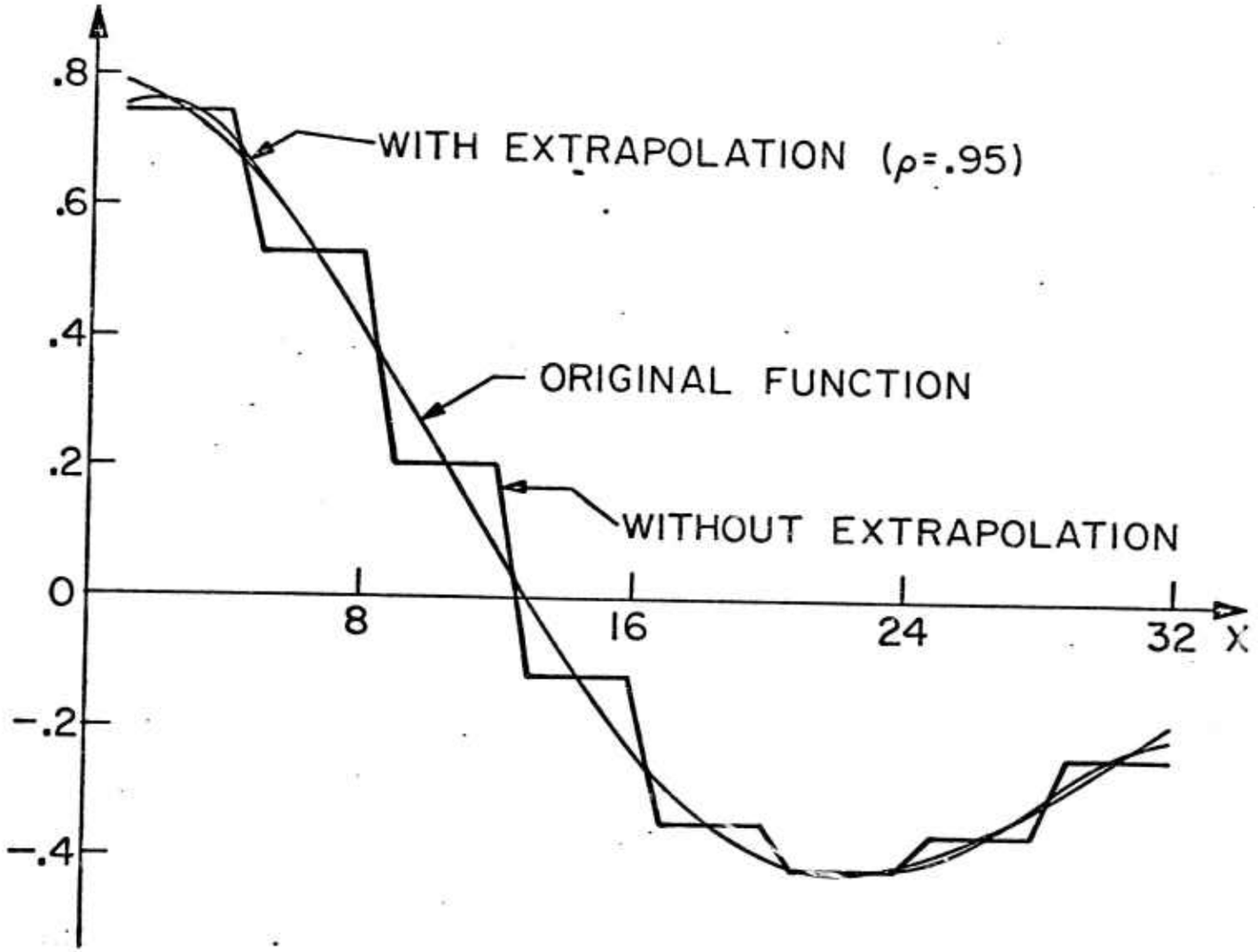
8. Experimental Results

Computer simulation experiments have been performed to evaluate the spectrum extrapolation process. Figure 3 contains an example of spectrum extrapolation in the Hadamard transform domain for two one-dimensional signals: a sin θ/θ waveform and a pulse superimposed on a quadratic background. Figure 4 illustrates two dimensional spectrum extrapolation in the Hadamard domain. In this example a 256x256 pixel image was transformed in 16x16 pixel bocks. The transform coefficients lying outside of 8x8 and 5x5 low sequency zones, for the two experiments, were truncated. The improvement in subjective quality is apparent from the photographs.

9. Summary

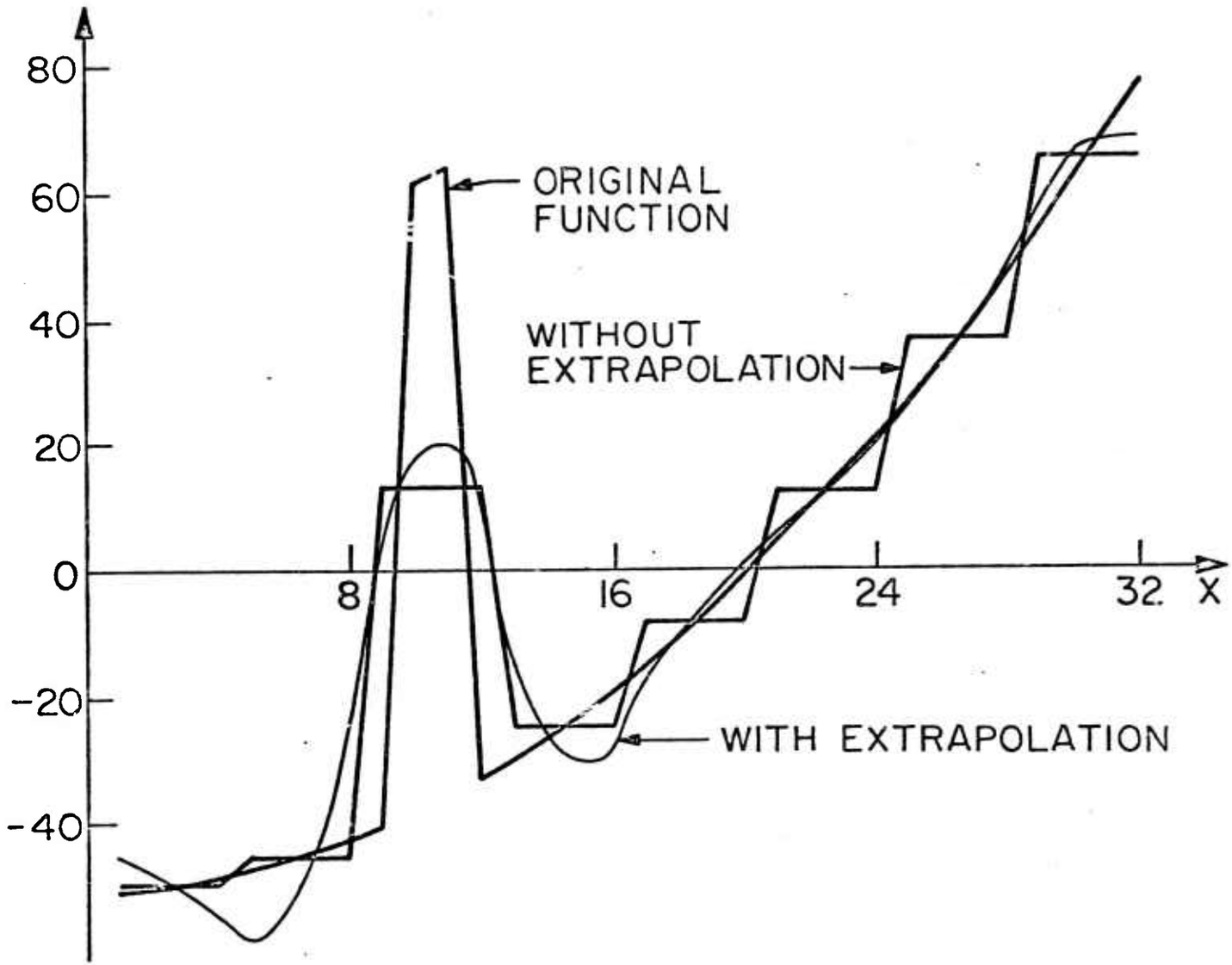
A relatively simple technique for extrapolating the values of missing transform coefficients in transform signal processing systems has been demonstrated. The technique possesses the interesting feature that it is an entirely aposteriori method. Thus, in a transform data coding system it is possible to obtain an increase in data reconstruction accuracy by processing at the decoder, without affecting the structure of the coder. Thus, in a data coding system, it may be worthwhile to utilize a transform with a relatively poor energy compaction property, but one possessing an efficient computational algorithm or implementation technology, and still achieve a desired level of system performance by spectrum extrapolation at the decoder.

Several interesting extensions of the spectrum extrapolation process remain to be explored. First, in many applications there is a priori knowledge available as to the structure of the reconstructed data which should be



(a) sin θ / θ function

Figure 3. Hadamard transform spectrum extrapolation of one dimensional functions for N = 32 and M = 8.



(b) Pulse on quadratic background function.

Figure 3. (continued)

used in the estimation of transform components. For example, in image coding it is known that the image reconstruction cannot possess negative points or points of extremely large magnitude. A second extension involves discovery of a method for interpolating the values of quantized transform coefficients for digital transform data coding systems in which gross quantization is applied to achieve a bandwidth compression. Finally, the development of recursive estimation algorithms to be used in conjunction with pipeline transform generators would be of interest.

Acknowledgement

Mr. Michael Huhns of the University of Southern California Electrical Engineering Department has provided valuable assistance in the computational examples presented here. His assistance is greatly appreciated.

References

1. P. A. Wintz, "Transform Picture Coding," IEEE Transactions on Information Theory, Volume 60, No. 7, July 1972, pp. 809-820.

2. J. P. Burg, "Maximum Entropy Spectral Analysis," 37th Meeting Society of Exploration Geophysicists, Oklahoma City, Oklahoma, 1972.

3. H. C. Andrews and W. K. Pratt, "Television Bandwidth Reduction by Encoding Spatial Frequencies," Journal of the SMPTE, volume 77, No. 2, December 1968, pp. 1279-1281.

4. N. Ahmed, T. Natarajan, and K. R. Rao, "On Image Processing and a Discrete Cosine Transform," (to be published)

5. H. C. Andrews, J. Kane and W. K. Pratt, "Hadamard Transform Image Coding," Proceedings of the IEEE, volume 57, No. 1, January 1969.

6. H. C. Andrews, Computer Techniques in Image Processing, Academic Press, New York, 1970.

7. W. K. Pratt, L. R. Welch and W. H. Chen, "Slant Transform for Image Coding," Proceedings of the Symposium on Application of Walsh Functions, March 1972.

8. Ali Habibi and P. A. Wintz, "Image Coding by Linear Transformation and Block Quantizations," IEEE Transactions on Communication Technology, volume COM-19, No. 1, February 1971, pp. 50-63.

9. A. Papoulis, Probability, Random Variables and Stochastic Processes, McGraw-Hill, Inc., 1965.

10. F. A. Graybill, Introduction to Matrices with Applications in Statistics, Wadsworth Publishing Co., Inc., Belmont, California, 1969.

11. A. Albert, Regression and Moore-Penrose Pseudoinverse, Academic Press, New York, 1972.

APPENDIX F

SIGNAL PROCESSING INTERPRETER

SIGNAL PROCESSING INTERPRETER: ADDITIONS AND MODIFICATIONS

by

B. R. WOOD

J. M. SPEISER

Sensor and Information Technology Department

Naval Undersea Center
San Diego, California 92132

ABSTRACT

This note describes additions and modifications to a signal processing interpreter which was developed as part of the ARPA sponsored program for image transmission via spread spectrum links. The signal processing interpreter SPIN3 is an interactive program for use at a time-sharing demand terminal. It provides the user with the equivalent of a calculator designed to perform signal processing operations and provides the software equivalent of a large number of modules for breadboarding a complete signal processing system.

This interpreter is meant to be used by engineers and scientists who are familiar with signal processing, but who may have no knowledge of programming.

The signal processing interpreter is particularly useful for the rapid investigation of systems whose complexity precludes a complete analytic study, and whose utilization of new components may make hardware breadboarding undesirable because of cost and procurement time limitations.

The present note describes software additions and modifications generated during the period July through September 1973 and is meant to be used in conjunction with the Signal Processing Interpreter Preliminary Description and User's Guide [1].

NEW COMMANDS AND NEWLY IMPLEMENTED COMMANDS

quantizer

correct

noise generator

add

clear

clear x

roll up

roll down

print all commands

scale

reciprocal

real part

imag part

expand

decimate

right circular shift

MODIFICATIONS

1. A re-read has been provided when read errors occur while entering commands or inputting a vector. The effect of this is that noise on the telephone link or user format errors will now only result in an error message and a request to re-enter the command or data point.

2. Previous input/output inconsistencies due to differences between Fortran and Algol conventions have been eliminated.

    a. All vector indices are now consistent; they run from 0 to N-1. These indices will appear in listings of vectors, plotting, and inputting.

    b. The implementation of the command CONSTANT FUNCTION has been rewritten to use a comma as the delimiter between the real and imaginary parts. This convention is now standard for all complex inputs.

3. The output printing has been cleaned up to minimize unnecessary line feeds and printing of redundant information.

DETAILED DESCRIPTION OF THE NEW COMMANDS AND NEWLY IMPLEMENTED COMMANDS

QUANTIZER - requests the user to enter an integer. The x vector is scaled so that the largest absolute value of the real or imaginary part of x is set equal to the specified integer. The real and imaginary parts are then rounded to the nearest integer.

CORRECT - permits the user to change the value of a single component of a vector. The user is asked to specify the index of the point to be changed, and is then asked to specify the new value in the format real part, imaginary part.

NOISE GENERATOR - generates a zero-mean random vector using a multiplicative congruential method described by Kruska [2]. The user supplies a value for the desired variance and an integer in the range 0-8191 for the desired initialization point.

ADD - pads the shorter of the X and Y vectors with zeros to the length of the longer one. The X vector is replaced by the sum of X and Y, and the stack is lowered.

CLEAR - sets all vectors to zero, with all dimensions set equal to 1.

CLEAR X - sets the X vector equal to zero, and sets the X dimension equal to one.

ROLL UP - performs a cyclic permutation of the vectors X, Y, Z, T and a similar permutation of their dimensions NX, NY, NZ, NT. Y is replaced by the

old value of X, etc.

ROLL DOWN - performs the inverse of ROLL UP.

PRINT ALL COMMANDS - causes the computer to print back each user-entered command which is recognized. This command is used primarily in running batch jobs, to provide a printout which is readable without reference to the card deck which was used for submission of the job.

SCALE - requests the user to enter a complex number. The X vector is multiplied by the number, and the result is placed in X.

RECIPROCAL - replaces every point in the X vector by its reciprocal.

REAL PART - replaces the X vector by its real part.

IMAG PART - replaces the X vector by its imaginary part, i.e., 1.2,3.4 is replaced by 3.4,0.0.

EXPAND - increases the length of the X vector by inserting zeros between the original data points. The user is requested to enter an integer for the expansion factor. If the integer k is entered, then k-1 zeros are placed between each pair of data points and after the last point.

DECIMATE - decreases the length of the X vector by resampling. The user is requested to enter a positive integer for the decimation factor. If the integer k is entered, then every kth point is saved.

RIGHT CIRCULAR SHIFT - produces a circular shift of the X vector. The user is asked to enter an integer for the shift value. If the integer is positive, it causes a right circular shift. If the integer is negative, it causes a left

circular shift. Since a circular shift by a multiple of NX is equivalent to a shift of zero, a circular shift of -k is equivalent to a circular shift of NX-k, so the distinction between right circular shifts and left circular shifts is purely conceptual.

REFERENCES

1. Speiser, J. M., Signal Processing Interpreter Preliminary Description and User's Guide, NUC TN 1065, 25 June 1973.

2. Kruskal, J. B., Extremely Portable Random Number Generator, Communications of the ACM, Volume 12, Number 2, Feb. 1969, pp. 93-94.

APPENDIX G

SAW ROW-TO-COLUMN CONVERSION

SIGNAL TRANSFORMATION WITH RECIRCULATING SAW DELAY LINES

by

C. M. Fortunko and H. J. Shaw

Pre-print

M. L. Report No. 2179

Revised July 1973

Contract No. N00014-67-A-0112-0039

submitted to

IEEE Transactions on

Sonics and Ultrasonics

Microwave Laboratory
W. W. Hansen Laboratories of Physics
Stanford University
Stanford, California

SIGNAL TRANSFORMATION WITH RECIRCULATING SAW DELAY LINES*

by

C. M. Fortunko and H. J. Shaw

Stanford University
Stanford, California

ABSTRACT

A system is described for transforming data by switching it into a bank of recirculating delay lines in one time sequence and switching it out in another sequence. This is applied to row-to-column conversion when the input data is in the form of an N × N matrix. An experimental demonstration is shown which uses SAW wrap-around delay line channels, transforming a 3 × 3 matrix of rf pulses of 16 μsec length and 52 MHz carrier frequency.

*The work reported in this paper was supported by the U.S. Office of Naval Research under the Joint Service Electronics Program by Contract N00014-67-A-0112-0039.

SIGNAL TRANSFORMATION WITH RECIRCULATING SAW DELAY LINES

by

C. M. Fortunko and H. J. Shaw

I. INTRODUCTION

Data processing in real time can be accomplished by storing incoming data in banks of recirculating surface wave delay lines and then reading out these data in such a way as to transform them in a desired fashion. We consider here some of the properties of such systems and describe the operation of a simple prototype system, operating at 52 MHz carrier frequency, which performs an operation equivalent to interchanging the rows and columns of a matrix.

The basic approach to the use of delay lines for such purposes is illustrated in Fig. 1. The essential item is a bank of identical recirculating surface wave delay line loops. Input data fed in at the input terminal of the system are distributed among the delay lines in some desired fashion by means of the WRITE circuitry. The data enter the delay lines through interdigital transducers indicated as arrows in the figure. The READ circuitry extracts the data from the delay lines by switching from one delay line to another in a time sequence which is different from that used for writing, to accomplish the required transformation of the data train.

The row-to-column converter can be used to perform the intermediate serial-to-parallel conversion which is required to implement decomposable two-dimensional transforms (e.g. Fourier, Hadamard). It can also be used to interface two system with different speeds.

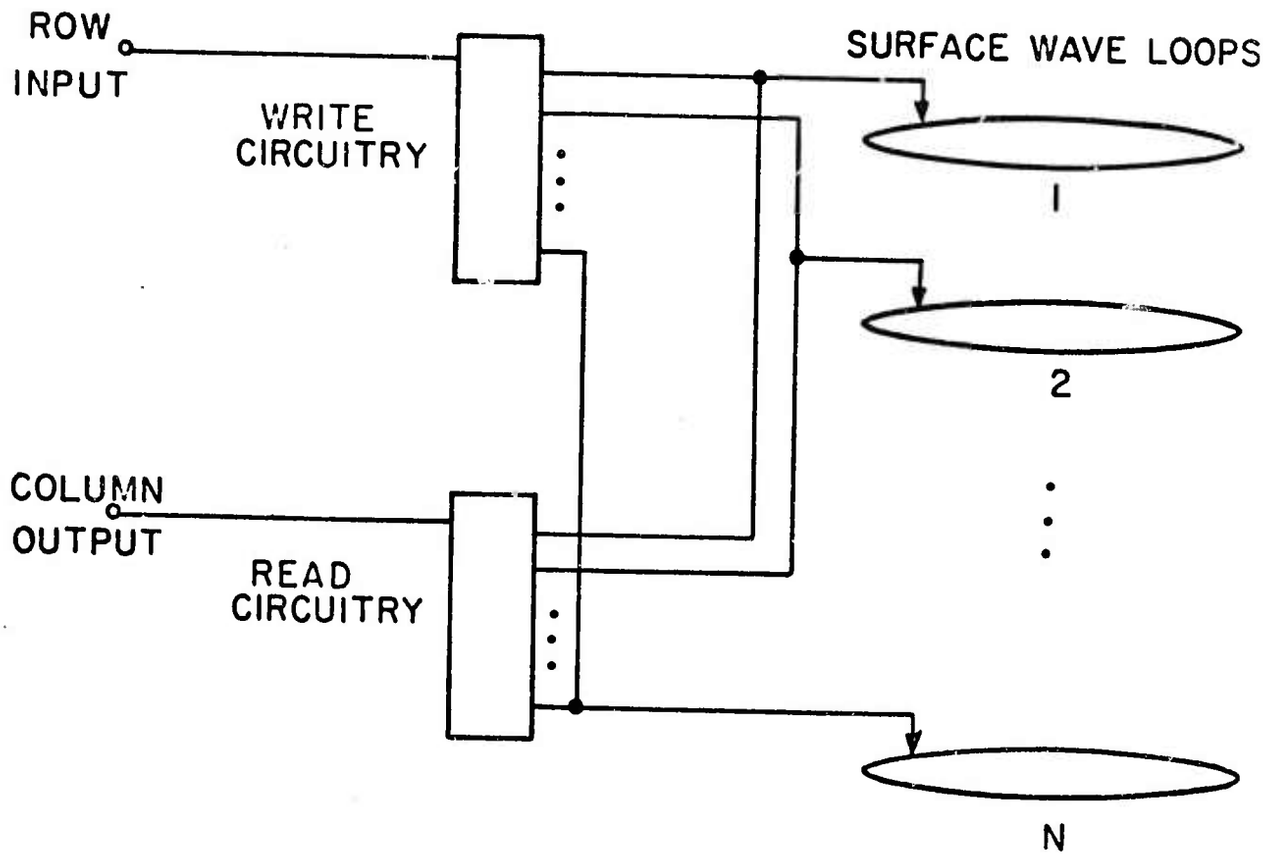


Fig. 1--Block diagram of data transformation system using closed-loop surface acoustic wave delay lines.

II. ROW-TO-COLUMN CONVERSION USING CLOSED-LOOP SURFACE ACOUSTIC WAVE DELAY LINES

We consider now an example of data transformation. Let the input data stream be $A_{11} \cdots A_{1N} \cdots A_{N1} \cdots A_{NN}$, which is ordered according to rows of the $N \times N$ matrix in Fig. 2. We now show that it is possible to devise WRITE and READ algorithms which will convert this into a sequence ordered by columns of the matrix, in real time, using only a single transducer on each delay line.

The transducers of Fig. 1 can be addressed at either of two rates, once per row or once per bit. To obtain the row-to-column conversion one of these rates is applied to the WRITE function and the other to the READ function. Either rate may be applied to either function giving two algorithms for the row-to-column operation. Figures 3(a) and (b) illustrate these two algorithms. In these diagrams successive horizontal lines show the signals seen by the transducer in successive delay lines as a function of time reading to the right. The squares represent the signals introduced into the transducers during the WRITE operation, and the quantities between curved brackets are the signals extracted from the transducers in the READ operation. The dashes indicate time slots containing no signal.

Figure 3(a) is for the case in which the WRITE function is performed at the row rate and the READ function at the bit rate. Each row of Fig. 2 is fed intact into a separate delay line. As indicated by the diagonal arrays of curved brackets, the sequence of gating on the transducers for readout is a uniform, periodic scan, 1, 2 . . . N , 1,2 . . . N, . . ., sequentially from line 1 to line N, in repetitive cycles. The final output

signal is shown at the bottom of Fig. 3(a). This output sequence is a real-time sequence free of gaps, ordered according to columns (time reversed) of the matrix of Fig. 2.

Figure 3(b) represents the alternate algorithm in which the WRITE and READ switching rates have been interchanged. For any row of input data of Fig. 2, the first bit enters the first storage loop, the second bit enters the second loop, and so forth until the entire row has been stored, one bit in each delay line. During the READ cycle the entire contents of each delay line are sampled, one row at a time. The sequence (column output) shown at the bottom of Fig. 3(b) is seen to be the same as for Fig. 3(a), ordered according to the columns (time reversed) of the matrix of Fig. 2. We see that the algorithms of Figs. 3(a) and (b) result in output sequences whose rows are columns of the input matrix as desired. However, both the rows and the elements within these rows appear in the opposite time sequence to the corresponding columns of Fig. 2. That is, the output sequences in Figs. 3(a) and (b) are time reversed. For cases of direct concern here, this is not of consequence. If it is not desired, however, it can be corrected by time-reversing the input signal by rows before feeding into the delay line loops. Figure 3(c) shows this behavior. The signals within each row have been time-reversed before insertion into the delay line loops. This time reversal could be done in a device having a time delay of N clock intervals, rather than $N \times N$ as would be required if row-to-row time reversal were also involved. A parametric acoustic time reversal could be employed for this operation.

Fig. 2--Matrix Notation.

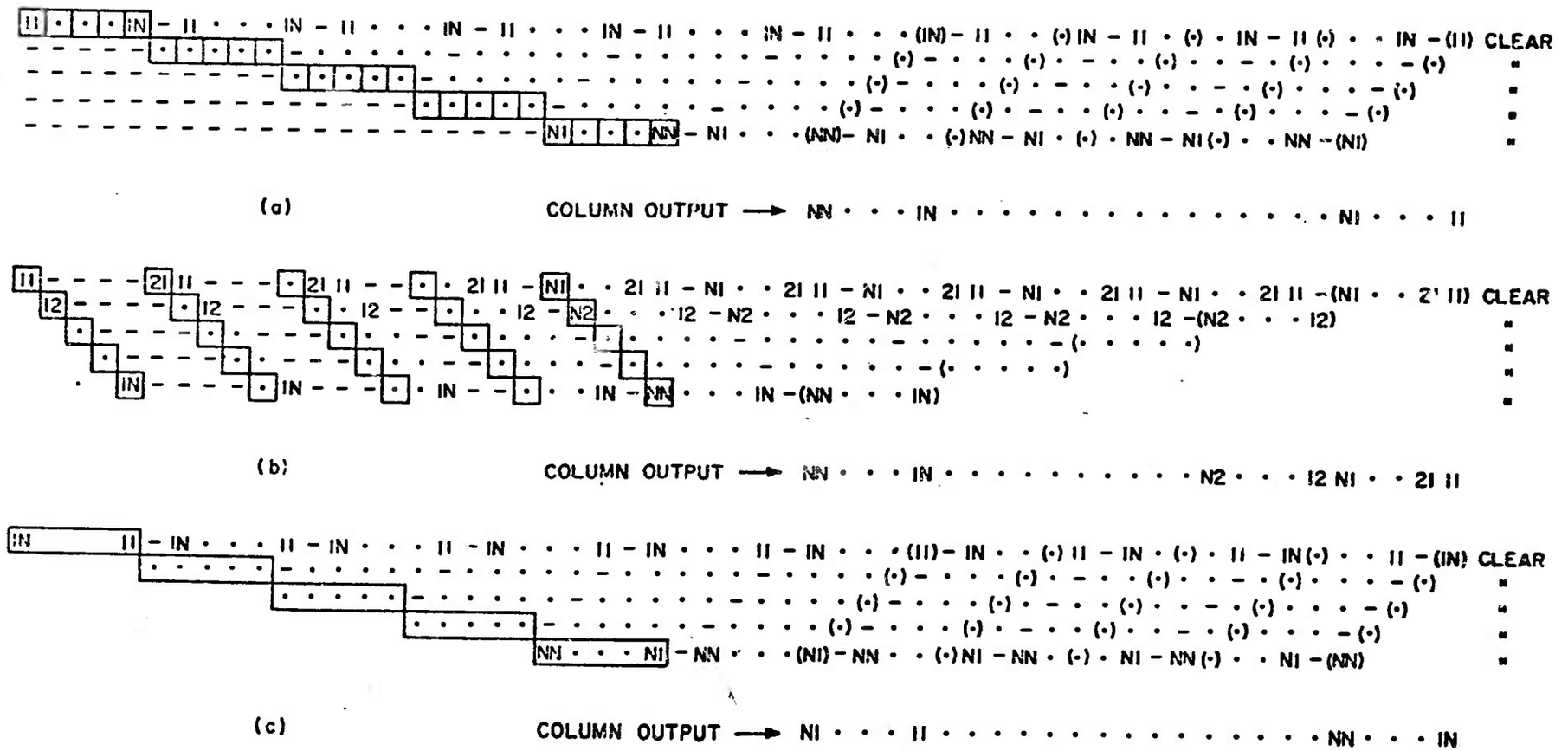


Fig. 3--Gating sequences for row-to-column conversion.